Noname manuscript No.
(will be inserted by the editor)

Entropy Stable h/p-Nonconforming Discretization with the Summation-by-Parts Property for the Compressible Euler and Navier–Stokes Equations

David C. Del Rey Fernández · Mark H. Carpenter · Lisandro Dalcin · Stefano Zampini · Matteo Parsani

Received: date / Accepted: date

Abstract In this paper, we extend the entropy conservative/stable algorithms presented by Del Rey Fernández and coauthors [20,19] for the compressible Euler and Navier–Stokes equations on nonconforming p-refined/coarsened curvilinear grids to h/p refinement/coarsening. The main difficulty in developing nonconforming algorithms is the construction of appropriate coupling procedures across nonconforming interfaces. Here, we utilize a computationally simple and efficient approach based upon using decoupled interpolation operators. The resulting scheme is entropy conservative/stable and elementwise conservative. Numerical simulations of the isentropic vortex and viscous shock propagation confirm the entropy conservation/stability and accuracy properties of the method (achieving $\sim p+1$ convergence), which are comparable to those of the original conforming scheme [5,42]. Simulations of the Taylor–Green vortex at $Re = 1,600$ and turbulent flow past a sphere at $Re_\infty = 2,000$ show the robustness and stability properties of the overall spatial discretization for unstructured grids. Finally, to demonstrate the entropy conservation property of a fully-discrete explicit entropy stable algorithm with h/p refinement/coarsening, we present the time evolution of the entropy function obtained by simulating the propagation of the isentropic vortex using a relaxation Runge–Kutta scheme.

David C. Del Rey Fernández
National Institute of Aerospace and Computational AeroSciences Branch, NASA Langley Research Center, Hampton, VA, United States
E-mail: dcdelrey@gmail.com

Mark H. Carpenter
NASA Langley Research Center, Hampton, VA, United States E-mail: mark.h.carpenter@nasa.gov

Lisandro Dalcin, Stefano Zampini, Matteo Parsani
King Abdullah University of Science and Technology (KAUST), Computer Electrical and Mathematical Science and Engineering Division (CEMSE), Extreme Computing Research Center (ECRC), Thuwal, Saudi Arabia E-mail: dalcinl@gmail.com, stefano.zampini@kaust.edu.sa, matteo.parsani@kaust.edu.sa

Keywords Nonconforming interfaces · h/p adaptation · Nonlinear entropy stability · Summation-by-parts · Simultaneous-approximation-terms · High-order accurate discretizations · Curved elements · Unstructured grid

1 Introduction

This paper is the final installment in a set of studies aimed at developing arbitrarily high-order, entropy stable, h/p-nonconforming schemes on curvilinear coordinates for the compressible Euler and Navier–Stokes equations [29,20,18,19]. The efficient use of exascale concurrency on next generation hardware motivates the search for algorithms that are accurate and robust. Moreover, essential to efficiency is the ability to optimally use degrees of freedom through h-, p-, and r-refinement/coarsening and communication hiding through dense compute kernels. High-order methods are natural candidates for next generation hardware because they are accurate and their ratio of communications to local computations is usually small; see, for instance, Refs. [32,33,2]. However, they have historically been limited by robustness issues, which is even more important as problem size and physics complexity increases.

When numerically solving partial differential equations (PDEs) it is imperative to find a bound on the growth rate of the solution; otherwise, the possibility exists that the solution could grow arbitrarily fast. This upper bound can be established by ensuring that a numerical method is stable. For linear variable coefficient problems in arbitrary dimensions, a general and systematic approach to ensure stability is the energy method because it can be applied to the continuous as well as the semidiscrete model. The energy method becomes extremely powerful when it is used in combination with the summation-by-parts (SBP) framework [22, 54] since it allows for the construction of provably stable schemes of any order. SBP operators can be viewed as strong or weak form differentiation matrices that mimic integration-by-parts (IBP) and are endowed with a telescoping property, critical for provable stability. Via this property, SBP schemes (augmented with appropriate interface coupling procedures, e.g., simultaneous approximation terms (SATs) [7,8,38,39,9,55,43,44]) reproduce, in a one-to-one manner, continuous stability proofs. Therefore, they provide a road map for the development of provably stable semidiscrete or fully discrete algorithms (see, for instance, Refs. [46,28]).

For nonlinear problems, a general and systematic approach for establishing stability has yet to be found. Nevertheless, for a certain class of PDEs, progress has been made. For conservation laws, Tadmor [56] constructed entropy conservative/stable low-order finite volume schemes that achieve entropy conservation by using two-point flux functions. When contracted with entropy variables, these schemes telescope the entropy flux. Entropy stability is then achieved by adding appropriate dissipation. For a review of these ideas see, for instance, Tadmor [57].

Tadmor's approach was extended to finite domains and arbitrary high-order finite difference WENO schemes in the work of Fisher and coauthors, who combined the SBP framework with Tadmor's two-point flux functions, resulting in entropy stable semidiscrete schemes [24,26,25]. This approach inherits all of the mechanics of linear SBP schemes for the imposition of boundary conditions and interelement coupling and gives a systematic methodology for discretizing problems on complex domains; see, for instance, Refs. [5,44,62,42,31,61,14,12,28,21]

and the references therein. An alternative method applicable to the compressible Euler equations [40,64,48,49], uses specially chosen entropy functions that result in a homogeneity property on the compressible Euler fluxes. Via this property, the Euler fluxes are split such that when contracted with the entropy variables, stability estimates result, which are analogous in form to energy estimates obtained for linear PDEs.

Until recently, the superior robustness and reliability of entropy stable discretizations were only investigated for inviscid flows [30,63,45] or low-speed viscous flows [27,35] in simple geometries. Recently, Rojas and coauthors [47] and Parsani and coauthors [41] have shown that low- and high-order entropy stable discontinuous spatial discretizations based on SBP–SAT operators provide an essential step toward a truly enabling technology in terms of reliability and robustness for underresolved turbulent flow simulations, flows with discontinuities, and flow past complex geometries.

The objective of this paper is the extension of entropy stable p-nonconforming algorithm presented in Del Rey Fernández et al. [20,18,19] for the compressible Euler and Navier–Stokes equations in curvilinear coordinates to arbitrary h/p-refinement/coarsening. The novel contributions of this paper are summarized as follows:

- A general and simple entropy conservative/stable nonconforming algorithm is proposed in curvilinear coordinates for the compressible Euler and Navier–Stokes equations that
  - Enables a simple extension of the algorithm in Refs. [20,18,19] that uses the same type of interface SAT and, therefore, allows code reutilization
  - Results in the solution of the discrete geometric conservation laws (GCL) that is local to each element
  - Applies the metric approximation approach of Crean et al. [14] to arbitrary h/p-nonconforming elements
  - Ensures free-stream preservation by satisfying the discrete GCL conditions
  - Is elementwise conservative
- Numerical evidence is provided to demonstrate that the scheme retains the stability and accuracy properties of the conforming base scheme [5,42]

The paper is organized as follows. The notation is summarized in Section 2. Next, and overview of the paper is given in Section 3. In Section 4, the h/p nonconforming algorithm is detailed in the context of the linear convection-diffusion equation. The required nonlinear mechanic necessary to extend the nonconforming algorithm to the compressible Navier–Stokes equations is described in the simple context of the Burgers' equation in Section 5. Section 6 details the extension of the nonconforming algorithm to the compressible Navier–Stokes equations. The addition of interface dissipation that retains the provable properties of the base algorithm is discussed in Section 7. Numerical experiments are detailed in Section 8, while conclusions are drawn in Section 9. To give further insight into the implementation of the algorithm, some key subroutines are provided in Appendix A.

2 Notation and definitions

The notation used herein is identical to that in Refs. [20,18,19]; readers familiar with the notation can skip to Section 4. PDEs are discretized on cubes having Cartesian computational coordinates denoted by the triple (ξ_1, ξ_2, ξ_3), where the physical coordinates are denoted by the triple (x_1, x_2, x_3). Vectors are represented by lowercase bold font, for example $\boldsymbol{u}$, while matrices are represented using sans-serif font, for example, B. Continuous functions on a space-time domain are denoted by capital letters in script font. For example,

$$\mathcal{U}(\xi_1, \xi_2, \xi_3, t) \in L^2([\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times [\alpha_3, \beta_3] \times [0, T])$$

represents a square integrable function, where t is the temporal coordinate. The restriction of such functions onto a set of mesh nodes is denoted by lower case bold font. For example, the restriction of $\mathcal{U}$ onto a grid of $N_1 \times N_2 \times N_3$ nodes is given by the vector

$$\boldsymbol{u} = \left[\mathcal{U}\left(\boldsymbol{\xi}^{(1)}, t\right), \ldots, \mathcal{U}\left(\boldsymbol{\xi}^{(N)}, t\right)\right]^{\mathrm{T}},$$

where, N is the total number of nodes ($N \equiv N_1 N_2 N_3$) square brackets ([]) are used to delineate vectors and matrices as well as ranges for variables (the context will make clear which meaning is being used). Moreover, $\boldsymbol{\xi}$ is a vector of vectors constructed from the three vectors $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$, and $\boldsymbol{\xi}_3$, which are vectors of size N_1, N_2, and N_3 and contain the coordinates of the mesh in the three computational directions, respectively. Finally, $\boldsymbol{\xi}$ is constructed as

$$\boldsymbol{\xi}(3(i-1)+1:3i) \equiv \boldsymbol{\xi}^{(i)} \equiv [\boldsymbol{\xi}_1(i), \boldsymbol{\xi}_2(i), \boldsymbol{\xi}_3(i)]^{\mathrm{T}},$$

where the notation $\boldsymbol{u}(i)$ means the i^{th} entry of the vector $\boldsymbol{u}$ and $\boldsymbol{u}(i:j)$ is the subvector constructed from $\boldsymbol{u}$ using the i^{th} through j^{th} entries (i.e., Matlab notation is used).

Oftentimes, monomials are discussed and the following notation is used:

$$\boldsymbol{\xi}_l^j \equiv \left[(\boldsymbol{\xi}_l(1))^j, \ldots, (\boldsymbol{\xi}_l(N_l))^j\right]^{\mathrm{T}},$$

and the convention that $\boldsymbol{\xi}_l^j = \boldsymbol{0}$ for $j < 0$ is used.

Herein, one-dimensional SBP operators are used to discretize derivatives. The definition of a one-dimensional SBP operator in the ξ_l direction, $l = 1, 2, 3$, can be found in Refs. [17,22,54]

Definition 1 Summation-by-parts operator for the first derivative: A matrix operator with constant coefficients, $\mathsf{D}_{\xi_l}^{(1D)} \in \mathbb{R}^{N_l \times N_l}$, is an SBP operator of degree p approximating the derivative $\frac{\partial}{\partial \xi_l}$ on the domain $\xi_l \in [\alpha_l, \beta_l]$ with nodal distribution $\boldsymbol{\xi}_l$ having N_l nodes, if

1. $\mathsf{D}_{\xi_l}^{(1D)} \boldsymbol{\xi}_l^j = j \boldsymbol{\xi}_l^{j-1}$, $j = 0, 1, \ldots, p$;
2. $\mathsf{D}_{\xi_l}^{(1D)} \equiv \left(\mathsf{P}_{\xi_l}^{(1D)}\right)^{-1} \mathsf{Q}_{\xi_l}^{(1D)}$, where the norm matrix, $\mathsf{P}_{\xi_l}^{(1D)}$, is symmetric positive definite;
3. $\mathsf{Q}_{\xi_l}^{(1D)} \equiv \left(\mathsf{S}_{\xi_l}^{(1D)} + \frac{1}{2}\mathsf{E}_{\xi_l}^{(1D)}\right)$, $\mathsf{S}_{\xi_l}^{(1D)} = -\left(\mathsf{S}_{\xi_l}^{(1D)}\right)^{\mathrm{T}}$, $\mathsf{E}_{\xi_l}^{(1D)} = \left(\mathsf{E}_{\xi_l}^{(1D)}\right)^{\mathrm{T}}$, $\mathsf{E}_{\xi_l}^{(1D)} = \mathrm{diag}(-1, 0, \ldots, 0, 1) = \boldsymbol{e}_{N_l}\boldsymbol{e}_{N_l}^{\mathrm{T}} - \boldsymbol{e}_{1_l}\boldsymbol{e}_{1_l}^{\mathrm{T}}$, $\boldsymbol{e}_{1_l} \equiv [1, 0, \ldots, 0]^{\mathrm{T}}$, and $\boldsymbol{e}_{N_l} \equiv [0, 0, \ldots, 1]^{\mathrm{T}}$.

Thus, a degree p SBP operator is one that differentiates exactly monomials up to degree p.

In this work, one-dimensional SBP operators are extended to multiple dimensions using tensor products ($\otimes$). The tensor product between the matrices A and B is given as $\mathsf{A} \otimes \mathsf{B}$. When referencing individual entries in a matrix the notation $\mathsf{A}(i, j)$ is used, which means the i^{th} j^{th} entry in the matrix A.

The focus in this paper is exclusively on diagonal-norm SBP operators. Moreover, the same one-dimensional SBP operators are used in each direction, each operating on N nodes. Specifically, diagonal-norm SBP operators constructed on the Legendre–Gauss–Lobatto (LGL) nodes are used, i.e., a discontinuous Galerkin collocated spectral element approach is utilized.

The physical domain $\Omega \subset \mathbb{R}^3$, with boundary $\Gamma \equiv \partial\Omega$ is partitioned into K nonoverlapping hexahedral elements. The domain of the κ^{th} element is denoted by Ω_κ and has boundary $\partial\Omega_\kappa$. Numerically, PDEs are solved in computational coordinates, where each Ω_κ is locally transformed to $\hat{\Omega}_\kappa$, with boundary $\hat{\Gamma} \equiv \partial\hat{\Omega}_\kappa$, under the following assumption:

Assumption 1 *Each element in physical space is transformed using a local and invertible curvilinear coordinate transformation that is compatible at shared interfaces, meaning that points in computational space on either side of a shared interface are mapped to the same physical location and, therefore, map back to the analogous location in computational space; this is the standard assumption that the curvilinear coordinate transformation is watertight.*

3 An overview of the paper

The required mechanics to construct entropy conservative/stable discretizations can be involved and this is made exponentially worse by the introduction of nonconforming interfaces. We have made an effort in this paper to help the reader in understanding both the theory as well as the practical implementation. As a result, there is a fair amount of material to digest. To help the reader further, in this section, we highlight the main ideas and give a guided tour of the paper.

At its core, the entropy conservative/stable discretization requires an SBP operator over the mesh and an appropriate two-point flux function. In other words, given a discretization that maintains the SBP property, an entropy conservative/stable discretization can always be constructed (assuming the continuous theory exists and an appropriate two-point flux function can be found). This notion allows us to separate the task of building an entropy conservative/stable discretization into 1) finding a discretization that maintains the SBP property for a linear analogue of the compressible Navier–Stokes equations and 2) extending this discretization to the compressible Navier–Stokes equations and combining it with an appropriate two-point flux function. This separation is also the approach taken in this paper.

The SBP property of elementwise SBP differentiation matrices is extended to the global SBP operator over the mesh by appropriate element coupling procedures, e.g., SATs. The main difficulty in constructing appropriate SATs for nonconforming interfaces is how to maintain the SBP property of the global mesh level SBP operator.

In Section 4, the construction of appropriate coupling procedures, obtained by interpolating between nonconforming meshes, is introduced in the context of the convection-diffusion equation (the linear analogue of the compressible Navier–Stokes equations that we use to develop our schemes). We use the notion of macro SBP operators over sets of nonconforming elements to elucidate the conditions required to obtain the SBP property for the global SBP operator. However, there are a multitude of ways of constructing appropriate interelement coupling and it is important that the final SBP operator decouples from the conditions required on the metric terms for free-stream preservation. The latter requirement leads to an algorithm that only necessitates local elementwise approximations to the metric terms so that the free-stream is preserved and entropy conservation/stability is achieved. The construction of SATs that satisfy this additional constraint is the main contribution of this paper. Moreover, in Section 4.4, we delineate how to approximate the metric terms appropriately.

With our nonconforming discretization that results in a global SBP operator, we then turn the attention to the required mechanics needed to construct an entropy conservative/stable discretization. Since the procedure is involved, we first focus on the simple context of the viscous Burgers' equation; Section 5. Next, in Section 6, we combine the nonconforming linear algorithm and the nonlinear mechanics to construct a discretization for the compressible Navier–Stokes equations. To further guide the reader on how to practically implement the discretization, we give additional details in Appendix A and Appendix B. The remainder of the paper is as described in the introduction.

4 An h/p-nonconforming algorithm: Linear convection-diffusion equation

In this paper, the focus is on curvilinearly mapped elements with interfaces that 1) are conforming but have nonconforming nodal distributions, such as would arise in p-refinement, 2) elements that have nonconforming faces, such as would arise in h-refinement, and 3) arbitrary combinations of 1) and 2). The development of an entropy stable h/p-refinement algorithm for the compressible Euler equations on Cartesian grids is detailed in Ref. [29]. The extension to curvilinear coordinates and p-refinement for the compressible Euler and Navier–Stokes equations is detailed in the series of papers [20,19,18], where an interface coupling technique is introduced that maintains, accuracy, discrete entropy conservation/stability and elementwise conservation and requires only local solves to approximate metric terms. Herein, the algorithm in Refs. [20,19,18] is extended to allow for arbitrary h/p-refinement on unstructured grids for the compressible Euler and Navier–Stokes equations.

4.1 Continuous and semidiscrete analysis

A number of key technical difficulties that arise in developing a stable and conservative nonconforming discretization for the compressible Navier–Stokes equations are already present in the simpler context of the linear convection-diffusion equation. As a result, the proposed interface coupling procedure for both the inviscid

and viscous terms are first presented for this simple linear scalar equation. In Cartesian coordinates, the linear convection-diffusion equation reads

$$\frac{\partial \mathcal{U}}{\partial t} + \sum_{m=1}^{3} \frac{\partial \left(a_m \mathcal{U}\right)}{\partial x_m} = \sum_{m=1}^{3} \frac{\partial^2 (b_m \mathcal{U})}{\partial x_m^2}, \tag{1}$$

where $(a_m \mathcal{U})$ and $\frac{\partial (b_m \mathcal{U})}{\partial x_m}$ are the inviscid and viscous fluxes, respectively. The symbols a_m correspond to the constant components of the convection speed whereas b_m are the positive and constant diffusion coefficients. The stability of (1) can be determined via the energy method, which proceeds by multiplying (1) by the solution, $(\mathcal{U})$, and after using the product rule yields

$$\frac{1}{2}\frac{\partial \mathcal{U}^2}{\partial t} + \frac{1}{2}\sum_{m=1}^{3} \frac{\partial \left(a_m \mathcal{U}^2\right)}{\partial x_m} = \sum_{m=1}^{3} \left\{ \frac{\partial}{\partial x_m}\left(\mathcal{U}\frac{\partial (b_m \mathcal{U})}{\partial x_m}\right) - \left(\frac{\partial (\sqrt{b_m}\mathcal{U})}{\partial x_m}\right)^2 \right\}. \tag{2}$$

Integrating over the domain, Ω, using integration by parts, and the Leibniz rule gives

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega} \frac{\mathcal{U}^2}{2} \mathrm{d}\Omega = \\ &\frac{1}{2}\sum_{m=1}^{3} \left(\oint_{\Gamma} \left\{ -\left(a_m \mathcal{U}^2\right) + 2\mathcal{U}\frac{(b_m \mathcal{U})}{\partial x_m} \right\} n_{x_m} \mathrm{d}\Gamma - 2\int_{\Omega} \left(\frac{\partial (\sqrt{b_m}\mathcal{U})}{\partial x_m}\right)^2 \mathrm{d}\Omega \right), \end{aligned} \tag{3}$$

where n_{x_m} is the m^{th} component of the outward facing unit normal. What Eq. (3) demonstrates is that the time rate of change of the norm of the solution, $\|\mathcal{U}\|^2 \equiv \int_{\Omega} \mathcal{U}^2 \mathrm{d}\Omega$, depends on surface flux integrals and a viscous dissipation term. This implies that, in combination with appropriate boundary conditions, Eq. (3) results in a bound on the solution in terms of the data of the problem, and, therefore, a proof of stability. The SBP framework used in this paper is mimetic of the above energy stability analysis in a one-to-one fashion and results in similar stability statements for the semidiscrete equations.

Derivatives are approximated using differentiation matrices that are defined in computational space. To do so, Eq. (1) is transformed using the curvilinear coordinate transformation $x_m = x_m(\xi_1, \xi_2, \xi_3)$. Thus, on the κ^{th} element, the x_m derivatives are expanded using the chain rule as

$$\frac{\partial}{\partial x_m} = \sum_{l=1}^{3} \frac{\partial \xi_l}{\partial x_m}\frac{\partial}{\partial \xi_l}, \qquad \frac{\partial^2}{\partial x_m^2} = \sum_{l,a=1}^{3} \frac{\partial \xi_l}{\partial x_m}\frac{\partial}{\partial \xi_l}\left(\frac{\partial \xi_a}{\partial x_m}\frac{\partial}{\partial \xi_a}\right).$$

Multiplying by the metric Jacobian, $(\mathcal{J}_\kappa)$, Eq. (1) becomes

$$\mathcal{J}_\kappa \frac{\partial \mathcal{U}}{\partial t} + \sum_{l,m=1}^{3} \mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\frac{\partial \left(a_m \mathcal{U}\right)}{\partial \xi_l} = \sum_{l,a,m=1}^{3} \mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\frac{\partial}{\partial \xi_l}\left(\frac{\partial \xi_a}{\partial x_m}\frac{\partial (b_m \mathcal{U})}{\partial \xi_a}\right). \tag{4}$$

Herein, Eq. (4) is referenced as the chain rule form of Eq. (1). Bringing the metric terms, $\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}$, inside the derivative and using the product rule gives

$$\begin{aligned} &\mathcal{J}_\kappa \frac{\partial \mathcal{U}}{\partial t} + \sum_{l,m=1}^{3} \frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m} a_m \mathcal{U}\right) - \sum_{l,m=1}^{3} a_m \mathcal{U} \frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\right) = \\ &\sum_{l,a,m=1}^{3} \frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\frac{\partial \xi_a}{\partial x_m}\frac{\partial (b_m \mathcal{U})}{\partial \xi_a}\right) - \sum_{l,a,m=1}^{3} \frac{\partial \xi_a}{\partial x_m}\frac{\partial (b_m \mathcal{U})}{\partial \xi_a}\frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\right). \end{aligned} \tag{5}$$

The last terms on the left- and right-hand sides of (5) are zero via the GCL relations

$$\sum_{l=1}^{3} \frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\right) = 0, \quad m = 1, 2, 3, \tag{6}$$

leading to the strong conservation form of the convection-diffusion equation in curvilinear coordinates:

$$\mathcal{J}_\kappa \frac{\partial \mathcal{U}}{\partial t} + \sum_{l,m=1}^{3} \frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m} a_m \mathcal{U}\right) = \sum_{l,a,m=1}^{3} \frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\frac{\partial \xi_a}{\partial x_m}\frac{\partial (b_m \mathcal{U})}{\partial \xi_a}\right). \tag{7}$$

The h-refinement procedure proceeds by subdividing the computational domain of parent elements, where children elements inherit the curvilinear coordinate transformation of the parent element. It is therefore convenient to introduce two computational coordinates:

- ξ_l, which is the mapping from the child element to physical space, i.e., the computational coordinate system of the κ^{th} element,
- $\hat{\xi}_l$, which is the mapping from the parent element to the physical space.

The mapping from children to parent elements is rectilinear. Thus, assuming that the child element has a computational domain of $[-1, 1]^3$, this transformation for the κ^{th} element is given by

$$\xi_l = \frac{2}{\Delta_l^\kappa}\hat{\xi}_l - \frac{\left(\hat{\mathrm{H}}_\kappa^l + \hat{\mathrm{L}}_\kappa^l\right)}{\Delta_l^\kappa}, \quad \Delta_l^\kappa \equiv \hat{\mathrm{H}}_\kappa^l - \hat{\mathrm{L}}_\kappa^l, \quad l = 1, 2, 3, \tag{8}$$

where $\hat{\mathrm{H}}_\kappa^l$ and $\hat{\mathrm{L}}_\kappa^l$ are the largest and smallest extent of the ξ_l coordinate in the coordinate system of the parent element ($\hat{\xi}_l$). Using Eq. (8) the Jacobian and metrics are recast in terms of the Jacobian, $\hat{\mathcal{J}}_\kappa$, and metrics terms, $\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}$, of the parent element. This step results in

$$\frac{\partial \xi_l}{\partial x_m} = \frac{2}{\Delta_l^\kappa}\frac{\partial \hat{\xi}_l}{\partial x_m}, \qquad \mathcal{J}_\kappa = \frac{\Delta_1^\kappa \Delta_2^\kappa \Delta_3^\kappa}{8}\hat{\mathcal{J}}_\kappa, \qquad \mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m} = \frac{\Delta_1^\kappa \Delta_2^\kappa \Delta_3^\kappa}{4\Delta_l^\kappa}\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}. \tag{9}$$

Inserting Eq. (9) into Eq. (4) and multiplying by $8/\left(\Delta_1^\kappa \Delta_2^\kappa \Delta_3^\kappa\right)$ gives

$$\hat{\mathcal{J}}_\kappa \frac{\partial \mathcal{U}}{\partial t} + \sum_{l,m=1}^{3} \frac{2}{\Delta_l^\kappa}\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\frac{\partial \left(a_m \mathcal{U}\right)}{\partial \xi_l} = \sum_{l,a,m=1}^{3} \frac{4}{\Delta_l^\kappa \Delta_a^\kappa}\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\left(\frac{\partial \hat{\xi}_a}{\partial x_m}\frac{\partial (b_m \mathcal{U})}{\partial \xi_a}\right). \tag{10}$$

Similarly, Eq. (5) is transformed to

$$\hat{\mathcal{J}}_\kappa \frac{\partial \mathcal{U}}{\partial t} + \sum_{l,m=1}^{3} \frac{2}{\Delta_l^\kappa} \frac{\partial}{\partial \xi_l} \left(\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} a_m \mathcal{U} \right) - \sum_{l,m=1}^{3} \frac{2}{\Delta_l^\kappa} a_m \mathcal{U} \frac{\partial}{\partial \xi_l} \left(\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right) =$$
$$\sum_{l,a,m=1}^{3} \frac{4}{\Delta_l^\kappa \Delta_a^\kappa} \frac{\partial}{\partial \xi_l} \left(\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \frac{\partial \hat{\xi}_a}{\partial x_m} \frac{\partial (b_m \mathcal{U})}{\partial \xi_a} \right)$$
$$- \sum_{l,a,m=1}^{3} \frac{4}{\Delta_l^\kappa \Delta_a^\kappa} \frac{\partial \hat{\xi}_a}{\partial x_m} \frac{\partial (b_m \mathcal{U})}{\partial \xi_a} \frac{\partial}{\partial \xi_l} \left(\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right), \tag{11}$$

and Eq. (7) is transformed to

$$\hat{\mathcal{J}}_\kappa \frac{\partial \mathcal{U}}{\partial t} + \sum_{l,m=1}^{3} \frac{2}{\Delta_l^\kappa} \frac{\partial}{\partial \xi_l} \left(\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} a_m \mathcal{U} \right) =$$
$$\sum_{l,a,m=1}^{3} \frac{4}{\Delta_l^\kappa \Delta_a^\kappa} \frac{\partial}{\partial \xi_l} \left(\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \frac{\partial \hat{\xi}_a}{\partial x_m} \frac{\partial (b_m \mathcal{U})}{\partial \xi_a} \right). \tag{12}$$

Directly discretizing Eq. (12) leads to semidiscrete schemes that are not guaranteed to be stable. Instead, a well-known approach is to use a canonical splitting of the inviscid terms that is constructed by using one half of the inviscid terms in Eq. (10) and one half of the inviscid terms in Eq. (11) (see, for instance Ref. [10]). On the other hand, the viscous terms are treated in strong conservation form. This process results in

$$\hat{\mathcal{J}}_\kappa \frac{\partial \mathcal{U}}{\partial t} + \frac{1}{2} \sum_{l,m=1}^{3} \frac{2}{\Delta_l^\kappa} \left\{ \frac{\partial}{\partial \xi_l} \left(\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} a_m \mathcal{U} \right) + \hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \frac{\partial}{\partial \xi_l} (a_m \mathcal{U}) \right\}$$
$$- \frac{1}{2} \sum_{l,m=1}^{3} \left\{ a_m \mathcal{U} \frac{\partial}{\partial \xi_l} \left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m} \right) \right\} = \sum_{l,a,m=1}^{3} \frac{4}{\Delta_l^\kappa \Delta_a^\kappa} \frac{\partial}{\partial \xi_l} \left(\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \frac{\partial \hat{\xi}_a}{\partial x_m} \frac{\partial (b_m \mathcal{U})}{\partial \xi_a} \right), \tag{13}$$

where the last set of terms on the left-hand side are zero by the GCL conditions (6). Now, consider discretizing Eq. (13) by using the following differentiation matrices:

$$\mathsf{D}_{\xi_1} \equiv \mathsf{D}^{(1D)} \otimes \mathsf{I}_N \otimes \mathsf{I}_N, \quad \mathsf{D}_{\xi_2} \equiv \mathsf{I}_N \otimes \mathsf{D}^{(1D)} \otimes \mathsf{I}_N, \quad \mathsf{D}_{\xi_3} \equiv \mathsf{I}_N \otimes \mathsf{I}_N \otimes \mathsf{D}^{(1D)},$$

where I_N is an $N \times N$ identity matrix. The diagonal matrices containing the metric Jacobian and metric terms along their diagonals, respectively, are defined as follows:

$$\hat{\mathsf{J}}_\kappa \equiv \mathrm{diag} \left(\hat{\mathcal{J}}_\kappa \left(\boldsymbol{\xi}^{(1)} \right), \ldots, \hat{\mathcal{J}}_\kappa \left(\boldsymbol{\xi}^{(N_\kappa)} \right) \right),$$
$$\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_\kappa \equiv \mathrm{diag} \left(\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} (\boldsymbol{\xi}^{(1)}), \ldots, \hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} (\boldsymbol{\xi}^{(N_\kappa)}) \right),$$

where $N_\kappa \equiv N^3$ is the total number of nodes in the element κ. With these matrices, the discretization of (13) on the κ^{th} element is given as

$$\begin{aligned}
&\hat{\mathsf{J}}_\kappa \frac{\mathrm{d}\boldsymbol{u}_\kappa}{\mathrm{d}t} + \frac{1}{2}\sum_{l,m=1}^{3} \frac{2}{\Delta_l^\kappa} a_m \left\{ \mathsf{D}_{\xi_l} \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_\kappa + \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_\kappa \mathsf{D}_{\xi_l} \right\} \boldsymbol{u}_\kappa \\
&- \frac{1}{2}\sum_{l,m=1}^{3} \left\{ \frac{2}{\Delta_l^\kappa} a_m \operatorname{diag}(\boldsymbol{u}_\kappa) \mathsf{D}_{\xi_l} \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_\kappa \mathbf{1}_\kappa \right\} = \\
&\sum_{l,m,a=1}^{3} \frac{4}{\Delta_l^\kappa \Delta_a^\kappa} b_m \mathsf{D}_{\xi_l}^\kappa \hat{\mathsf{J}}_\kappa^{-1} \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_\kappa \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_a}{\partial x_m} \right]_\kappa \mathsf{D}_{\xi_a}^\kappa \boldsymbol{u}_\kappa,
\end{aligned} \tag{14}$$

where $\mathbf{1}_\kappa$ is a vector of ones of the size of the number of nodes on the κ^{th} element and the SATs have been dropped as they are not important for the current analysis. In the same way as in the continuous case, the semidiscrete equations have an associated set of discrete GCL conditions

$$\sum_{l=1}^{3} \frac{2}{\Delta_l^\kappa} \mathsf{D}_{\xi_l} \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_\kappa \mathbf{1}_\kappa = \mathbf{0}, \quad m = 1, 2, 3, \tag{15}$$

that if satisfied, lead to the following telescoping and therefore, provably stable semidiscrete form:

$$\begin{aligned}
&\hat{\mathsf{J}}_\kappa \frac{\mathrm{d}\boldsymbol{u}_\kappa}{\mathrm{d}t} + \frac{1}{2}\sum_{l,m=1}^{3} \frac{2}{\Delta_l^\kappa} a_m \left\{ \mathsf{D}_{\xi_l} \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_\kappa + \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_\kappa \mathsf{D}_{\xi_l} \right\} \boldsymbol{u}_\kappa = \\
&\sum_{l,m,a=1}^{3} \frac{4}{\Delta_l^\kappa \Delta_a^\kappa} b_m \mathsf{D}_{\xi_l}^\kappa \hat{\mathsf{J}}_\kappa^{-1} \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_\kappa \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_a}{\partial x_m} \right]_\kappa \mathsf{D}_{\xi_a}^\kappa \boldsymbol{u}_\kappa.
\end{aligned} \tag{16}$$

How to construct metrics that satisfy the discrete GCL conditions of Eq. (15) will be detailed later in the paper and is one of the major contributions of this work. In the next subsection, attention is focused on the construction of appropriate interface coupling procedures that retain the stability (telescoping) properties of the scheme represented by Eq. (16) across h/p nonconforming elements (the pure p nonconforming case is detailed in Refs. [19,20,18]).

4.2 The nonconforming interface

To simplify both the analysis as well as the presentation of the semidiscrete scheme discussed in this paper, it is convenient to focus the attention on a shared interface between two elements only, one of which is h/p-refined. Without loss of generality, five elements are considered that have aligned computational coordinates and adjoin along a vertical interface; see Fig. 1. The focus is on nonconformities that arise from both h refinement/coarsening as well as local approximations with differing polynomial degrees, as would result from p-refinement/coarsening. Thus, the generic example, Fig. 1, considers a left element having polynomial degree p_{L} and a set of four right elements having polynomial degrees p_{R_f}, $f = 1, 2, 3, 4$, possibly all having differing degrees and not equal to p_{L} (i.e., they originate from

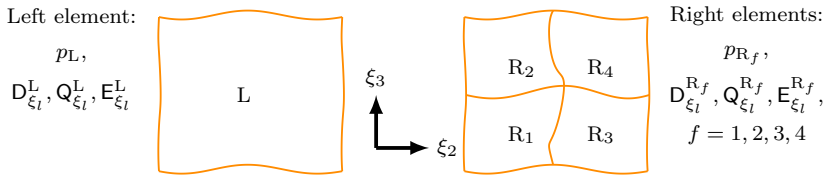


Fig. 1: Generic surface used to describe the construction of various quantities as well as to simplify the analysis of the semidiscrete schemes.

a conforming right element that has been h-refined and then p-refined/coarsened). Therefore, the contributions from the left element are identified with subscript or superscript L, and similarly for the right elements with subscripts or superscripts R_f; see Fig. 1.

The analysis proceeds by developing macro matrix differentiation operators over the five elements, i.e., composed of elements L and R_f, $f = 1, 2, 3, 4$, and then determining the required modifications/restrictions so that the resulting operators have the SBP property. A naive construction, in the computational coordinates of the parent elements, would be the following operators:

$$\widetilde{\mathsf{D}}_{\hat{\xi}_l} \equiv \begin{bmatrix} \mathsf{D}^{\mathrm{L}}_{\hat{\xi}_l} & & & \\ & \mathsf{D}^{\mathrm{R}_1}_{\hat{\xi}_l} & & \\ & & \ddots & \\ & & & \mathsf{D}^{\mathrm{R}_4}_{\hat{\xi}_l} \end{bmatrix}, \quad l = 1, 2, 3, \tag{17}$$

where the elementwise components of the SBP operators, for example $\mathsf{D}^{\mathrm{L}}_{\hat{\xi}_1}$, are constructed as

$$\mathsf{D}^{\mathrm{L}}_{\hat{\xi}_1} \equiv \frac{2}{\Delta^{\mathrm{L}}_1} \mathsf{D}_{\xi_1}, \quad \mathsf{D}_{\xi_1} \equiv \mathsf{D}^{(1D)}_{\mathrm{L}} \otimes \mathsf{I}_{\mathrm{L}} \otimes \mathsf{I}_{\mathrm{L}},$$

$$\mathsf{P}_{\mathrm{L}} \equiv \frac{\Delta^{\mathrm{L}}_1 \Delta^{\mathrm{L}}_2 \Delta^{\mathrm{L}}_3}{8} \mathsf{P}^{(1D)}_{\mathrm{L}} \otimes \mathsf{P}^{(1D)}_{\mathrm{L}} \otimes \mathsf{P}^{(1D)}_{\mathrm{L}},$$

$$\mathsf{Q}^{\mathrm{L}}_{\hat{\xi}_1} \equiv \frac{\Delta^{\mathrm{L}}_2 \Delta^{\mathrm{L}}_3}{4} \mathsf{Q}^{(1D)}_{\mathrm{L}} \otimes \mathsf{P}^{(1D)}_{\mathrm{L}} \otimes \mathsf{P}^{(1D)}_{\mathrm{L}},$$

$$\mathsf{E}^{\mathrm{L}}_{\hat{\xi}_1} \equiv \frac{\Delta^{\mathrm{L}}_2 \Delta^{\mathrm{L}}_3}{4} \left\{ \boldsymbol{e}^{\mathrm{L}}_N \left(\boldsymbol{e}^{\mathrm{L}}_N \right) \otimes \mathsf{P}^{(1D)}_{\mathrm{L}} \otimes \mathsf{P}^{(1D)}_{\mathrm{L}} - \boldsymbol{e}^{\mathrm{L}}_1 \left(\boldsymbol{e}^{\mathrm{L}}_1 \right) \otimes \mathsf{P}^{(1D)}_{\mathrm{L}} \otimes \mathsf{P}^{(1D)}_{\mathrm{L}} \right\},$$

where I_{L} is an identity matrix of size $N^{1/3}_{\mathrm{L}} \times N^{1/3}_{\mathrm{L}}$ and N_{L} is the total number of nodes in element L.

The macro element operators $\widetilde{\mathsf{D}}_{\hat{\xi}_i}$ are not SBP (i.e., they do not telescope to the boundaries). To make them SBP appropriate, the interelement coupling needs to be introduced. For the $\widetilde{\mathsf{D}}_{\hat{\xi}_2}$ and $\widetilde{\mathsf{D}}_{\hat{\xi}_3}$ operators, this is achieved using standard SATs. For the $\widetilde{\mathsf{D}}_{\hat{\xi}_1}$ operator, interpolation operators that interpolate information from elements R_f to element L and vice versa are needed. For simplicity, the interpolation operators use only tensor product surface information from the adjoining interface surface.

With this background, general matrix difference operators between the five elements are constructed as

$$\widetilde{\mathsf{D}}_{\hat{\xi}_l} = \widetilde{\mathsf{P}}^{-1}\widetilde{\mathsf{Q}}_{\hat{\xi}_l} = \widetilde{\mathsf{P}}^{-1}\left(\widetilde{\mathsf{S}}_{\hat{\xi}_l} + \frac{1}{2}\widetilde{\mathsf{E}}_{\hat{\xi}_l}\right). \tag{18}$$

Focusing on the direction orthogonal to the interface (ξ_1), the relevant matrices are given by

$$\begin{aligned}
&\widetilde{\mathsf{P}} \equiv \mathrm{diag}\left(\mathsf{P}_{\mathrm{L}}, \mathsf{P}_{\mathrm{R}_1}, \mathsf{P}_{\mathrm{R}_2}, \mathsf{P}_{\mathrm{R}_3}, \mathsf{P}_{\mathrm{R}_4}\right),\\
&\widetilde{\mathsf{S}}_{\hat{\xi}_1} \equiv \begin{bmatrix} \mathsf{S}^{\mathrm{L}}_{\hat{\xi}_1} & \tilde{\mathsf{S}}_{12} & \tilde{\mathsf{S}}_{13} & \tilde{\mathsf{S}}_{14} & \tilde{\mathsf{S}}_{15} \\ \tilde{\mathsf{S}}_{21} & \mathsf{S}^{\mathrm{R}_1}_{\hat{\xi}_1} & & & \\ \tilde{\mathsf{S}}_{31} & & \mathsf{S}^{\mathrm{R}_2}_{\hat{\xi}_1} & & \\ \tilde{\mathsf{S}}_{41} & & & \mathsf{S}^{\mathrm{R}_3}_{\hat{\xi}_1} & \\ \tilde{\mathsf{S}}_{51} & & & & \mathsf{S}^{\mathrm{R}_4}_{\hat{\xi}_1} \end{bmatrix},\\
&\tilde{\mathsf{S}}_{12} \equiv \frac{\Delta_2^{\mathrm{L}}\Delta_3^{\mathrm{L}}}{4}\boldsymbol{e}_N^{\mathrm{L}}\left(\boldsymbol{e}_1^{\mathrm{R}_1}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^2_{\mathrm{R}_1\mathrm{toL}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^3_{\mathrm{R}_1\mathrm{toL}},\\
&\tilde{\mathsf{S}}_{21} \equiv -\frac{\Delta_2^{\mathrm{R}_1}\Delta_3^{\mathrm{R}_1}}{4}\boldsymbol{e}_1^{\mathrm{R}_1}\left(\boldsymbol{e}_N^{\mathrm{L}}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{R}_1}^{(1D)}\mathsf{I}^2_{\mathrm{LtoR}_1} \otimes \mathsf{P}_{\mathrm{R}_1}^{(1D)}\mathsf{I}^3_{\mathrm{LtoR}_1},\\
&\tilde{\mathsf{S}}_{13} \equiv \frac{\Delta_2^{\mathrm{L}}\Delta_3^{\mathrm{L}}}{4}\boldsymbol{e}_N^{\mathrm{L}}\left(\boldsymbol{e}_1^{\mathrm{R}_2}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^2_{\mathrm{R}_2\mathrm{toL}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^3_{\mathrm{R}_2\mathrm{toL}},\\
&\tilde{\mathsf{S}}_{31} \equiv -\frac{\Delta_2^{\mathrm{R}_2}\Delta_3^{\mathrm{R}_2}}{4}\boldsymbol{e}_1^{\mathrm{R}_2}\left(\boldsymbol{e}_N^{\mathrm{L}}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{R}_2}^{(1D)}\mathsf{I}^2_{\mathrm{LtoR}_2} \otimes \mathsf{P}_{\mathrm{R}_2}^{(1D)}\mathsf{I}^3_{\mathrm{LtoR}_2},\\
&\tilde{\mathsf{S}}_{14} \equiv \frac{\Delta_2^{\mathrm{L}}\Delta_3^{\mathrm{L}}}{4}\boldsymbol{e}_N^{\mathrm{L}}\left(\boldsymbol{e}_1^{\mathrm{R}_3}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^2_{\mathrm{R}_3\mathrm{toL}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^3_{\mathrm{R}_3\mathrm{toL}},\\
&\tilde{\mathsf{S}}_{41} \equiv -\frac{\Delta_2^{\mathrm{R}_3}\Delta_3^{\mathrm{R}_3}}{4}\boldsymbol{e}_1^{\mathrm{R}_3}\left(\boldsymbol{e}_N^{\mathrm{L}}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{R}_3}^{(1D)}\mathsf{I}^2_{\mathrm{LtoR}_3} \otimes \mathsf{P}_{\mathrm{R}_3}^{(1D)}\mathsf{I}^3_{\mathrm{LtoR}_3},\\
&\tilde{\mathsf{S}}_{15} \equiv \frac{\Delta_2^{\mathrm{L}}\Delta_3^{\mathrm{L}}}{4}\boldsymbol{e}_N^{\mathrm{L}}\left(\boldsymbol{e}_1^{\mathrm{R}_4}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^2_{\mathrm{R}_4\mathrm{toL}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^3_{\mathrm{R}_5\mathrm{toL}},\\
&\tilde{\mathsf{S}}_{51} \equiv -\frac{\Delta_2^{\mathrm{L}}\Delta_3^{\mathrm{R}_4}}{4}\boldsymbol{e}_1^{\mathrm{R}_4}\left(\boldsymbol{e}_N^{\mathrm{L}}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{R}_4}^{(1D)}\mathsf{I}^2_{\mathrm{LtoR}_5} \otimes \mathsf{P}_{\mathrm{R}_5}^{(1D)}\mathsf{I}^3_{\mathrm{LtoR}_5},\\
&\widetilde{\mathsf{E}}_{\hat{\xi}_1} \equiv \begin{bmatrix} \tilde{\mathsf{E}}_{11} & & & \\ & \tilde{\mathsf{E}}_{22} & & \\ & & \ddots & \\ & & & \tilde{\mathsf{E}}_{55} \end{bmatrix},\\
&\tilde{\mathsf{E}}_{11} \equiv -\frac{\Delta_2^{\mathrm{L}}\Delta_3^{\mathrm{L}}}{4}\boldsymbol{e}_1^{\mathrm{L}}\left(\boldsymbol{e}_1^{\mathrm{L}}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)},\\
&\tilde{\mathsf{E}}_{22} \equiv \frac{\Delta_2^{\mathrm{R}_1}\Delta_3^{\mathrm{R}_1}}{4}\boldsymbol{e}_N^{\mathrm{R}_1}\left(\boldsymbol{e}_N^{\mathrm{R}_1}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{R}_1}^{(1D)} \otimes \mathsf{P}_{\mathrm{R}_1}^{(1D)},\\
&\tilde{\mathsf{E}}_{55} \equiv \frac{\Delta_2^{\mathrm{R}_4}\Delta_3^{\mathrm{R}_4}}{4}\boldsymbol{e}_N^{\mathrm{R}_4}\left(\boldsymbol{e}_N^{\mathrm{R}_4}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{R}_4}^{(1D)} \otimes \mathsf{P}_{\mathrm{R}_4}^{(1D)},
\end{aligned} \tag{19}$$

and $\mathsf{I}^l_{\mathrm{R}_f\mathrm{toL}}$ and $\mathsf{I}^l_{\mathrm{LtoR_f}}$ are one-dimensional interpolation operators, in the l direction, from the R_f element to the L element and vice versa.

A necessary constraint that the SBP formalism places on $\widetilde{\mathsf{D}}_{\hat{\xi}_1}$ is skew-symmetry of the $\widetilde{\mathsf{S}}_{\hat{\xi}_1}$ matrices. The block-diagonal matrices in $\widetilde{\mathsf{S}}_{\hat{\xi}_1}$ are already skew-symmetric but the off diagonal blocks are not. Thus, it is necessary to satisfy the following conditions:

$$\tilde{\mathsf{S}}_{12} = -\tilde{\mathsf{S}}_{21}^{\mathrm{T}}, \quad \tilde{\mathsf{S}}_{13} = -\tilde{\mathsf{S}}_{31}^{\mathrm{T}}, \quad \tilde{\mathsf{S}}_{14} = -\tilde{\mathsf{S}}_{41}^{\mathrm{T}}.$$

This implies that the interpolation operators are related to each other as follows:

$$\mathsf{I}^{l}_{\mathrm{R}_f\mathrm{toL}} = \frac{\Delta_l^{\mathrm{R}_f}}{\Delta_l^{\mathrm{L}}} \left(\mathsf{P}_{\mathrm{L}}^{(1D)}\right)^{-1} \left(\mathsf{I}^{l}_{\mathrm{LtoR_f}}\right)^{\mathrm{T}} \mathsf{P}_{\mathrm{R}_1}^{(1D)}, \quad l = 1, 2, \quad f = 1, 2, 3, 4.$$

This property is denoted as the SBP preserving property because it leads to a macro element differentiation matrix that is an SBP operator. The interpolation operators from the left element to the right elements is constructed using an L_2 projection approach such that

$$\mathsf{I}^{l}_{\mathrm{LtoR_f}} \equiv \left(\mathsf{M}^{\hat{\xi}_l}_{\mathrm{R}_f}\right)^{-1} \mathsf{M}^{\hat{\xi}_l}_{\mathrm{LtoR}_f},$$

$$\mathsf{M}^{\hat{\xi}_l}_{\mathrm{R}_f}(i,j) \equiv \int_{\hat{\mathrm{L}}^{l}_{\mathrm{R}_f}}^{\hat{\mathrm{H}}^{l}_{\mathrm{R}_f}} \mathcal{L}^{(i)}_{\hat{\xi}_l,\mathrm{R}_f} \mathcal{L}^{(j)}_{\hat{\xi}_l,\mathrm{R}_f} \mathrm{d}\hat{\xi}_l, \quad i,j = 1,\ldots,\left(p_{\mathrm{R}_f}+1\right),$$

$$\mathsf{M}^{\hat{\xi}_l}_{\mathrm{LtoR}_f}(i,j) \equiv \int_{\hat{\mathrm{L}}^{l}_{\mathrm{R}_f}}^{\hat{\mathrm{H}}^{l}_{\mathrm{R}_f}} \mathcal{L}^{(i)}_{\hat{\xi}_l,\mathrm{R}_f} \mathcal{L}^{(j)}_{\hat{\xi}_l,\mathrm{L}} \mathrm{d}\hat{\xi}_l, \quad i = 1,\ldots,\left(p_{\mathrm{R}_f}+1\right) \quad j = 1,\ldots,\left(p_{\mathrm{L}}+1\right),$$

where $\mathcal{L}^{(i)}_{\hat{\xi}_l,\mathrm{R}_f}$ and $\mathcal{L}^{(j)}_{\hat{\xi}_l,\mathrm{L}}$ are the i^{th} and j^{th} Lagrange basis functions constructed from the nodes of element R_f and L in the parent elements coordinates, respectively. Now theorems on the accuracy of the interpolation operators are presented

Theorem 1 *The interpolation operator* $\mathsf{I}^{l}_{\mathrm{LtoR_f}}$ *is of degree* $\min\left(N_{\mathrm{L}}^{1/3}-1, N_{\mathrm{R}_f}^{1/3}-1\right)$.

Proof The proof is standard and is not included for brevity. It follows by expanding out the matrices and taking advantage of the interpolating property of the Lagrangian basis functions (see, for example, Section 2.1.2 in Fredrich et al. [29] (a mortar is introduced but conceptually the proofs are identical).

The interpolation operators $\mathsf{I}^{l}_{\mathrm{R}_f\mathrm{toL}}$ individually are not polynomial exact, but rather, their combination is.

Theorem 2 *The combined interpolation from the right elements to the left element is of degree* $\min\left(N_{\mathrm{L}}-2, N_{\mathrm{R}_1}-2,\ldots,N_{\mathrm{R}_4}-2\right)$*, if the norms are suboptimal, i.e., degree* $2p-1$*, otherwise it is of degree* $\min\left(N_{\mathrm{L}}-1, N_{\mathrm{R}_1}-1,\ldots,N_{\mathrm{R}_4}-1\right)$*. In the five-element example used herein, the combined interpolation operator, acting on some function* $\mathcal{U}$*, is*

$$\mathsf{I}_{\mathrm{R}_1\mathrm{toL}}\boldsymbol{u}_{\mathrm{R}_1} + \mathsf{I}_{\mathrm{R}_2\mathrm{toL}}\boldsymbol{u}_{\mathrm{R}_2} + \mathsf{I}_{\mathrm{R}_3\mathrm{toL}}\boldsymbol{u}_{\mathrm{R}_3} + \mathsf{I}_{\mathrm{R}_4\mathrm{toL}}\boldsymbol{u}_{\mathrm{R}_4},$$

where $\boldsymbol{u}_{\mathrm{R}_f}$ *is the vector containing the evaluation of the function* $\mathcal{U}$ *at the nodes of the abutting surface of the* R_f *element and*

$$\mathsf{I}_{\mathrm{R}_f\mathrm{toL}} \equiv \mathsf{I}^{2}_{\mathrm{R}_f\mathrm{toL}} \otimes \mathsf{I}^{3}_{\mathrm{R}_f\mathrm{toL}}, \quad f = 1, 2, 3, 4.$$

Proof This proof follows in the same way as proven elsewhere, for example, see Ref. [29].

The semidiscrete skew-symmetric split operator given in Eq. (13), discretized using the macro element operators $\widetilde{\mathsf{D}}_{\hat{\xi}_l}$, and metric terms, $\hat{\mathsf{J}}$, $\left[\mathcal{J}\frac{\partial \xi_l}{\partial x_m}\right]$, leads to the following scheme:

$$\begin{aligned}\hat{\mathsf{J}}\frac{\mathrm{d}\widetilde{\boldsymbol{u}}}{\mathrm{d}t} + \frac{1}{2}\sum_{l,m=1}^{3} a_m \left(\widetilde{\mathsf{D}}_{\hat{\xi}_l}\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\right] + \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\right]\widetilde{\mathsf{D}}_{\hat{\xi}_l}\right)\widetilde{\boldsymbol{u}} \\ - \frac{1}{2}\sum_{l,m=1}^{3} a_m \,\mathrm{diag}\left(\widetilde{\boldsymbol{u}}\right)\widetilde{\mathsf{D}}_{\hat{\xi}_l}\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\right]\tilde{\mathbf{1}} = \sum_{l,a=1}^{3}\widetilde{\mathsf{D}}_{\hat{\xi}_l}\left[\hat{\mathsf{C}}_{l,a}\right]\widetilde{\mathsf{D}}_{\hat{\xi}_a}\widetilde{\boldsymbol{u}},\end{aligned} \tag{20}$$

where

$$\begin{aligned}&\widetilde{\boldsymbol{u}} \equiv \left[\boldsymbol{u}_{\mathrm{L}}^{\mathrm{T}}, \boldsymbol{u}_{\mathrm{R}_1}^{\mathrm{T}}, \boldsymbol{u}_{\mathrm{R}_2}^{\mathrm{T}}, \boldsymbol{u}_{\mathrm{R}_3}^{\mathrm{T}}, \boldsymbol{u}_{\mathrm{R}_4}^{\mathrm{T}}\right]^{\mathrm{T}}, \hat{\mathsf{J}} \equiv \mathrm{diag}\begin{bmatrix}\hat{\mathsf{J}}_{\mathrm{L}} & & & \\ & \hat{\mathsf{J}}_{\mathrm{R}_1} & & \\ & & \ddots & \\ & & & \hat{\mathsf{J}}_{\mathrm{R}_4}\end{bmatrix}, \\ &\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\right] \equiv \begin{bmatrix}\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\right]_{\mathrm{L}} & & & \\ & \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\right]_{\mathrm{R}_1} & & \\ & & \ddots & \\ & & & \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\right]_{\mathrm{R}_4}\end{bmatrix}, \\ &\left[\hat{\mathsf{C}}_{l,a}\right] \equiv \sum_{m=1}^{3} b_m \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\right] = \begin{bmatrix}\left[\hat{\mathsf{C}}_{l,a}\right]_{\mathrm{L}} & & & \\ & \left[\hat{\mathsf{C}}_{l,m}\right]_{\mathrm{R}_1} & & \\ & & \ddots & \\ & & & \left[\hat{\mathsf{C}}_{l,m}\right]_{\mathrm{R}_4}\end{bmatrix}.\end{aligned} \tag{21}$$

As was the case in Eq. (14), a necessary condition for stability is that the metric terms satisfy the following discrete GCL conditions:

$$\sum_{l=1}^{3}\widetilde{\mathsf{D}}_{\hat{\xi}_l}\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\right]\tilde{\mathbf{1}} = \mathbf{0}. \tag{22}$$

Unfortunately, since $\widetilde{\mathsf{D}}_{\hat{\xi}_1}$ is not a tensor product operator and therefore, does not commute with the other derivative matrix operators, discrete metrics constructed using the analytic formalism of Vinokur and Yee [59] or Thomas and Lombard [58] will not in general satisfy the discrete GCL condition required in Eq. (22). This means that, instead, the metric terms have to be constructed so that they directly satisfy the GCL constraints.

Remark 1 The metric terms are assigned colors; e.g., the time-term Jacobian: $\hat{\mathsf{J}}$ or the volume metric terms: $\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\right]$. Metric terms with common colors form

a set that must be computed consistently. For example, the time-term Jacobian and the volume metric Jacobian may not be computed in the same way. Another important set of metrics are the surface metrics, which are introduced in the next subsection.

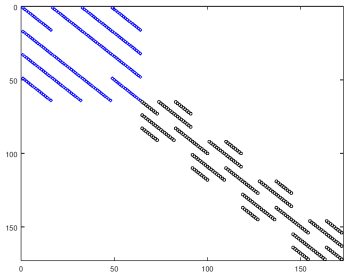
4.3 Isolating the metric terms

The discrete GCL system, Eq. (22) is highly underdetermined and couples the approximation of the metric terms in all five elements. In general, the resulting GCL conditions for arbitrary h/p-refinement would couple large sets of elements, making the solution of Eq. (22) difficult if not impossible. Note that the GCL conditions originate from the spatial discretization of the skew-symmetric splitting of the convective terms. Thus, if the approximation for those terms can be appropriately modified, then a set of element-local discrete GCL conditions can be constructed, making the problem tractable again. This is precisely the approach taken in Refs. [20,19,18] in the context of p-refinement/coarsening, and it is the same procedure used herein.

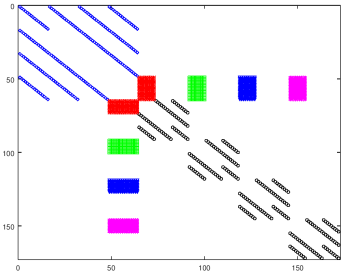
Examining the volume terms for the approximation of the skew-symmetric splitting highlights how to decouple the discrete GCL conditions:

$$
\begin{aligned}
&\widetilde{\mathsf{P}}\left(\widetilde{\mathsf{D}}_{\hat{\xi}_1}\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m}\right] + \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m}\right]\widetilde{\mathsf{D}}_{\hat{\xi}_1}\right) = \\
&\begin{bmatrix}
\mathsf{A}_{11} & \mathsf{A}_{12} & \mathsf{A}_{13} & \mathsf{A}_{14} & \mathsf{A}_{15} \\
-\mathsf{A}_{12}^{\mathrm{T}} & \mathsf{A}_{22} & & & \\
-\mathsf{A}_{13}^{\mathrm{T}} & & \mathsf{A}_{33} & & \\
-\mathsf{A}_{14}^{\mathrm{T}} & & & \mathsf{A}_{44} & \\
-\mathsf{A}_{15}^{\mathrm{T}} & & & & \mathsf{A}_{55}
\end{bmatrix} + \left(\widetilde{\mathsf{E}}_{\hat{\xi}_1}\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m}\right] + \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m}\right]\widetilde{\mathsf{E}}_{\hat{\xi}_1}\right), \\
&\mathsf{A}_{11} \equiv \left\{\mathsf{S}^{\mathrm{L}}_{\hat{\xi}_1}\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m}\right]_{\mathrm{L}} + \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m}\right]_{\mathrm{L}}\mathsf{S}^{\mathrm{L}}_{\hat{\xi}_1}\right\}, \\
&\mathsf{A}_{1f} = \frac{\Delta_2^{\mathrm{L}}\Delta_3^{\mathrm{L}}}{4}\left\{\begin{array}{l}
\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m}\right]_{\mathrm{L}}\left(\boldsymbol{e}_N^{\mathrm{L}}\left(\boldsymbol{e}_1^{\mathrm{R}_f}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^2_{\mathrm{R}_f\mathrm{toL}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^3_{\mathrm{R}_f\mathrm{toL}}\right) \\
+\left(\boldsymbol{e}_N^{\mathrm{L}}\left(\boldsymbol{e}_1^{\mathrm{R}_f}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^2_{\mathrm{R}_f\mathrm{toL}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^3_{\mathrm{R}_f\mathrm{toL}}\right)\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m}\right]_{\mathrm{R}_f}
\end{array}\right\}, \\
&\mathsf{A}_{ff} \equiv \left\{\mathsf{S}^{\mathrm{R}_f}_{\hat{\xi}_1}\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m}\right]_{\mathrm{R}_f} + \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m}\right]_{\mathrm{R}_f}\mathsf{S}^{\mathrm{R}_f}_{\hat{\xi}_1}\right\}, \quad f = 1,2,3,4.
\end{aligned}
\tag{23}
$$

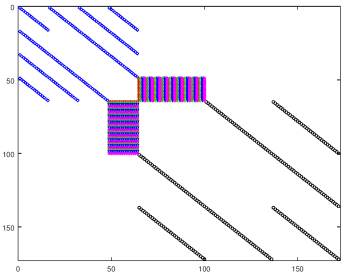
The highlighted terms are responsible for the weak coupling in the discrete GCL constraints. Note that these can be replaced with any design order quantities. The approach taken here to decouple the discrete GCL conditions is to zero the terms associated with the surface metrics on the element L, i.e., the terms $\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m}\right]_{\mathrm{L}}$ and to specify the terms on the R_f elements, i.e., $\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m}\right]_{\mathrm{R}_f}$.



(a) Original macro (non SBP) $\widetilde{\mathsf{D}}_{\hat{\xi}_1}$ operator without coupling.



(b) Modified SBP macro element operator $\widetilde{\mathsf{D}}_{\hat{\xi}_1}$.



(c) SBP macro element $\widetilde{\mathsf{D}}_{\hat{\xi}_1}$ organized in lexicographical order.

Fig. 2: Nonzero pattern of the various macro $\widetilde{\mathsf{D}}_{\hat{\xi}_1}$ operators.

Remark 2 In contrast to the p-adaptation case [19,20,18], we do not use surface metric terms from both sides of the element. This is because using surface metric terms from the L element results in a coupled system of equations for the GCL conditions in Eq. (22).

The action of the interface coupling is illustrated in Fig. 2. Using the above approach, the discrete GCL conditions, Eq. (22), become (where contributions from the boundary SATs have been ignored)

$$\begin{aligned}
&\sum_{l=1}^{3} \mathsf{D}_{\hat{\xi}_l}^{\mathrm{L}} \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_{\mathrm{L}} \mathbf{1}_{\mathrm{L}} = \\
&\frac{\Delta_2^{\mathrm{R}_f} \Delta_3^{\mathrm{R}_f}}{4} \left(\mathsf{P}^{\mathrm{L}} \right)^{-1} \left\{ \mathsf{R}_{\mathrm{L}}^{\mathrm{T}} \left(\mathsf{P}_{\mathrm{L}}^{(1D)} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)} \right) \mathsf{R}_{\mathrm{L}} \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m} \right]_{\mathrm{L}} \right\} \mathbf{1}_{\mathrm{L}} \\
&- \frac{\Delta_2^{\mathrm{R}_f} \Delta_3^{\mathrm{R}_f}}{4} \left(\mathsf{P}^{\mathrm{L}} \right)^{-1} \sum_{f=1}^{4} \left\{ \mathsf{R}_{\mathrm{L}}^{\mathrm{T}} \left(\mathsf{P}_{\mathrm{L}}^{(1D)} \mathsf{I}^2_{\mathrm{R}_f \mathrm{toL}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)} \mathsf{I}^3_{\mathrm{R}_f \mathrm{toL}} \right) \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m} \right]_{\mathrm{R}_f}^{\hat{\Gamma}} \mathsf{R}_{\mathrm{R}_f} \right\} \mathbf{1}_{\mathrm{R}_f},
\end{aligned} \tag{24}$$

$$\sum_{l=1}^{3} \mathsf{D}_{\hat{\xi}_l}^{\mathrm{R}_f} \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_{\mathrm{R}_f} \mathbf{1}_{\mathrm{R}_f} =$$
$$- \frac{\Delta_2^{\mathrm{R}_f} \Delta_3^{\mathrm{R}_f}}{4} \left(\mathsf{P}^{\mathrm{R}_f} \right)^{-1} \left\{ \mathsf{R}_{\mathrm{R}_f}^{\mathrm{T}} \left(\mathsf{P}_{\mathrm{R}_f}^{(1D)} \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \right) \mathsf{R}_{\mathrm{R}_f} \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m} \right]_{\mathrm{R}_f} \right\} \mathbf{1}_{\mathrm{R}_f}$$
$$+ \frac{\Delta_2^{\mathrm{R}_f} \Delta_3^{\mathrm{R}_f}}{4} \left(\mathsf{P}^{\mathrm{R}_f} \right)^{-1} \left\{ \mathsf{R}_{\mathrm{R}_f}^{\mathrm{T}} \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m} \right]_{\mathrm{R}_f}^{\hat{\Gamma}} \left(\mathsf{P}_{\mathrm{R}_f}^{(1D)} \mathsf{I}_{\mathrm{LtoR_f}}^2 \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \mathsf{I}_{\mathrm{LtoR_f}}^3 \right) \mathsf{R}_{\mathrm{L}} \right\} \mathbf{1}_{\mathrm{L}}, \tag{25}$$

where

$$\mathsf{R}_{\mathrm{L}} \equiv \left(\boldsymbol{e}_N^{\mathrm{L}} \right)^{\mathrm{T}} \otimes \mathsf{I}_{\mathrm{L}} \otimes \mathsf{I}_{\mathrm{L}}, \quad \mathsf{R}_{\mathrm{R}_f} \equiv \left(\boldsymbol{e}_1^{\mathrm{R}_f} \right)^{\mathrm{T}} \otimes \mathsf{I}_{\mathrm{R}_f} \otimes \mathsf{I}_{\mathrm{R}_f}.$$

The matrices $\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m} \right]_{\mathrm{R}_f}^{\hat{\Gamma}}$ are of size $N_{\mathrm{R}_f}^{2/3} \times N_{\mathrm{R}_f}^{2/3}$ and their diagonal elements are approximations to the metrics on the surface nodes of element R_f at the shared interface. In order to decouple the five systems of equations in Eqs. (24) and (25), the terms in $\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m} \right]_{\mathrm{R}_f}^{\hat{\Gamma}}$ need to be specified by, for example, using the analytic metrics, which is the approach taken in this paper. For later use, we introduce notation for the macro element $\widetilde{\mathsf{D}}_{l,m}$, which is the macro element operator constructed as described above for the metric terms $\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}$.

The next section reviews how to construct the metrics so that the discrete GCL conditions in Eqs. (24) and (25) are satisfied.

The presentation using the macro elements helps to simplify proofs and highlight the main issues involved in developing appropriate coupling procedures at the nonconforming interface. However, to give further insight into how to implement the algorithm in a computer code, here, we explicitly unroll the macro SBP operators into the discretization on the five elements involved at the interface (we give further details on implementation in Appendix A).

On the L element, the discretization reads as

$$
\begin{aligned}
&\hat{\mathsf{J}}_{\mathrm{L}}\frac{\mathrm{d}\boldsymbol{u}_{\mathrm{L}}}{\mathrm{d}t}+\frac{1}{2}\sum_{l,m=1}^{3}\left(\mathsf{D}^{\mathrm{L}}_{\hat{\xi}_l}\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_l}{\partial x_m}\right]_{\mathrm{L}}+\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_l}{\partial x_m}\right]_{\mathrm{L}}\mathsf{D}^{\mathrm{L}}_{\hat{\xi}_l}\right)\boldsymbol{u}_{\mathrm{L}}=\sum_{l,a=1}^{3}\mathsf{D}^{\mathrm{L}}_{\hat{\xi}_l}\left[\hat{\mathsf{C}}_{l,a}\right]_{\mathrm{L}}\boldsymbol{\theta}^{\mathrm{L}}_a\\
&+\mathsf{P}^{-1}_{\mathrm{L}}\sum_{m=1}^{3}\left\{\left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\right)\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_1}{\partial x_m}\right]_{\mathrm{L}}\boldsymbol{u}_{\mathrm{L}}\right.\\
&\qquad\left.-\left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\left(\boldsymbol{e}^{\mathrm{R}_f}_{1_1}\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\mathsf{I}^{2}_{\mathrm{R}_f\mathrm{toL}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\mathsf{I}^{3}_{\mathrm{R}_f\mathrm{toL}}\right)\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_1}{\partial x_m}\right]_{\mathrm{R}_f}\boldsymbol{u}_{\mathrm{R}_f}\right\}\\
&+\frac{1}{2}\mathsf{P}^{-1}_{\mathrm{L}}\sum_{a=1}^{3}\left\{\left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\right)\left[\hat{\mathsf{C}}_{1,a}\right]_{\mathrm{L}}\boldsymbol{\theta}^{\mathrm{L}}_a\right.\\
&\qquad\left.-\sum_{f=1}^{4}\left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\left(\boldsymbol{e}^{\mathrm{R}_f}_{1_1}\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\mathsf{I}^{2}_{\mathrm{R}_f\mathrm{toL}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\mathsf{I}^{3}_{\mathrm{R}_f\mathrm{toL}}\right)\left[\hat{\mathsf{C}}_{1,a}\right]_{\mathrm{R}_f}\boldsymbol{\theta}^{\mathrm{R}_f}_a\right\}\\
&+\boldsymbol{SAT}_{\mathrm{L}},
\end{aligned}
\tag{26}
$$

where $\boldsymbol{SAT}_{\mathrm{L}}$ contains the contributions from inviscid and viscous SATs on the remaining faces of element L. Moreover, the auxiliary variables $\boldsymbol{\theta}^{\mathrm{L}}_a$ and $\boldsymbol{\theta}^{\mathrm{R}_f}_a$ have been introduced. The first is constructed from

$$
\begin{aligned}
\boldsymbol{\theta}_{\mathrm{L}}=\mathsf{D}^{\mathrm{L}}_{\hat{\xi}_l}\boldsymbol{u}_{\mathrm{L}}&-\frac{1}{2}\mathsf{P}^{-1}_{\mathrm{L}}\sum_{a=1}^{3}\left\{\left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\right)\boldsymbol{u}_{\mathrm{L}}\right.\\
&\left.-\sum_{f=1}^{4}\left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\left(\boldsymbol{e}^{\mathrm{R}_f}_{1_1}\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\mathsf{I}^{2}_{\mathrm{R}_f\mathrm{toL}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\mathsf{I}^{3}_{\mathrm{R}_f\mathrm{toL}}\right)\boldsymbol{u}_{\mathrm{R}_f}\right\}.
\end{aligned}
\tag{27}
$$

The discretization on the R_f elements reads

$$
\begin{aligned}
&\hat{\mathsf{J}}_{\mathrm{R}_f}\frac{\mathrm{d}\boldsymbol{u}_{\mathrm{R}_f}}{\mathrm{d}t}+\frac{1}{2}\sum_{l,m=1}^{3}\left(\mathsf{D}^{\mathrm{R}_f}_{\hat{\xi}_l}\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_l}{\partial x_m}\right]_{\mathrm{R}_f}+\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_l}{\partial x_m}\right]_{\mathrm{R}_f}\mathsf{D}^{\mathrm{R}_f}_{\hat{\xi}_l}\right)\boldsymbol{u}_{\mathrm{R}_f}=\sum_{l,a=1}^{3}\mathsf{D}^{\mathrm{R}_f}_{\hat{\xi}_l}\left[\hat{\mathsf{C}}_{l,a}\right]_{\mathrm{R}_f}\boldsymbol{\theta}^{\mathrm{R}_f}_a\\
&-\mathsf{P}^{-1}_{\mathrm{R}_f}\sum_{m=1}^{3}\left\{\left(\boldsymbol{e}^{\mathrm{R}_f}_{1_1}\left(\boldsymbol{e}^{\mathrm{R}_f}_{1_1}\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\right)\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_1}{\partial x_m}\right]_{\mathrm{R}_f}\boldsymbol{u}_{\mathrm{R}_f}\right.\\
&\qquad\left.-\left(\boldsymbol{e}^{\mathrm{R}_f}_{1_1}\left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\mathsf{I}^{2}_{\mathrm{LtoR_f}}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\mathsf{I}^{3}_{\mathrm{LtoR_f}}\right)\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_1}{\partial x_m}\right]_{\mathrm{L}}\boldsymbol{u}_{\mathrm{L}}\right\}\\
&+\frac{1}{2}\mathsf{P}^{-1}_{\mathrm{R}_f}\sum_{a=1}^{3}\left\{\left(\boldsymbol{e}^{\mathrm{R}_f}_{1_1}\left(\boldsymbol{e}^{\mathrm{R}_f}_{1_1}\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\right)\left[\hat{\mathsf{C}}_{1,a}\right]_{\mathrm{R}_f}\boldsymbol{\theta}^{\mathrm{R}_f}_a\right.\\
&\qquad\left.-\left(\boldsymbol{e}^{\mathrm{R}_f}_{1_1}\left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\mathsf{I}^{2}_{\mathrm{LtoR_f}}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\mathsf{I}^{3}_{\mathrm{LtoR_f}}\right)\left[\hat{\mathsf{C}}_{1,a}\right]_{\mathrm{L}}\boldsymbol{\theta}^{\mathrm{L}}_a\right\}\\
&+\boldsymbol{SAT}_{\mathrm{R}_f},
\end{aligned}
\tag{28}
$$

where $\boldsymbol{SAT}_{\mathrm{R}_f}$ contains the contributions from inviscid and viscous SATs on the remaining faces of element R_f, and

$$\boldsymbol{\theta}_{\mathrm{R}_f} = \mathsf{D}^{\mathrm{R}_f}_{\hat{\xi}_l}\boldsymbol{u}_{\mathrm{R}_f} + \frac{1}{2}\mathsf{P}^{-1}_{\mathrm{R}_f}\sum_{a=1}^{3}\left\{\left(\boldsymbol{e}^{\mathrm{R}_f}_{1_1}\left(\boldsymbol{e}^{\mathrm{R}_f}_{1_1}\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\right)\boldsymbol{u}_{\mathrm{R}_f} - \left(\boldsymbol{e}^{\mathrm{R}_f}_{1_1}\left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\mathsf{I}^{2}_{\mathrm{LtoR_f}}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\mathsf{I}^{3}_{\mathrm{LtoR_f}}\right)\boldsymbol{u}_{\mathrm{L}}\right\}. \tag{29}$$

4.4 Metric solution mechanics

This section details the approximation of the metric terms so that entropy stability and free-stream preservation are maintained. There are two sets of metrics that need to be approximated, the volume metrics and the surface metrics. What needs to be satisfied are the discrete GCL equations (24) and (25), which are recast below in a form that is more convenient for developing a solution procedure. Thus, multiplying the discrete GCL constraints by -1, using the SBP property $\mathsf{Q} = -\mathsf{Q}^{\mathrm{T}} + \mathsf{E}$, and simplifying the expressions gives

$$\begin{aligned}&\sum_{l=1}^{3}\left(\mathsf{Q}^{\mathrm{L}}_{\hat{\xi}_l}\right)^{\mathrm{T}}\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_l}{\partial x_m}\right]_{\mathrm{L}}\mathbf{1}_{\mathrm{L}} = \\ &\frac{\Delta^{\mathrm{R}_f}_2\Delta^{\mathrm{R}_f}_3}{4}\sum_{f=1}^{4}\left\{\mathsf{R}^{\mathrm{T}}_{\mathrm{L}}\left(\mathsf{P}^{(1D)}_{\mathrm{L}}\mathsf{I}^{2}_{\mathrm{R}_f\mathrm{toL}}\otimes\mathsf{P}^{(1D)}_{\mathrm{L}}\mathsf{I}^{3}_{\mathrm{R}_f\mathrm{toL}}\right)\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_1}{\partial x_m}\right]^{\hat{\Gamma}}_{\mathrm{R}_f}\mathsf{R}_{\mathrm{R}_f}\right\}\mathbf{1}_{\mathrm{R}_f},\end{aligned} \tag{30}$$

$$\begin{aligned}&\sum_{l=1}^{3}\mathsf{Q}^{\mathrm{R}_f}_{\hat{\xi}_l}\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_l}{\partial x_m}\right]_{\mathrm{R}_f}\mathbf{1}_{\mathrm{R}_f} = \\ &-\frac{\Delta^{\mathrm{R}_f}_2\Delta^{\mathrm{R}_f}_3}{4}\left(\mathsf{P}^{\mathrm{R}_f}\right)^{-1}\left\{\mathsf{R}^{\mathrm{T}}_{\mathrm{R}_f}\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_1}{\partial x_m}\right]^{\hat{\Gamma}}_{\mathrm{R}_f}\left(\mathsf{P}^{(1D)}_{\mathrm{R}_f}\mathsf{I}^{2}_{\mathrm{LtoR_f}}\otimes\mathsf{P}^{(1D)}_{\mathrm{R}_f}\mathsf{I}^{3}_{\mathrm{LtoR_f}}\right)\mathsf{R}_{\mathrm{L}}\right\}\mathbf{1}_{\mathrm{L}},\end{aligned} \tag{31}$$

where $\mathsf{I}_{\mathrm{R}_f}$ is an identity matrix of size $N^{1/3}_{\mathrm{R}_f}\times N_{\mathrm{R}_f}$ and N_{R_f} is the total number of nodes in element R_f.

Note that the contributions from the $\mathsf{E}_{\hat{\xi}_l}$ from the left-hand side (i.e., coming from the step $\mathsf{Q} = -\mathsf{Q}^{\mathrm{T}} + \mathsf{E}$) related to the boundaries of the macro element are ignored. These contributions interact with the boundary SATs in the same way as the interface.

The metric terms in Eqs. (30) and (31) are set by solving a strictly convex quadratic optimization problem, based on the algorithm proposed in Crean et al. [14] (see also Refs. [20,18]). Here the procedure is exemplified in terms of the discrete GCL system on the L element:

$$\begin{aligned}&\min_{\boldsymbol{a}^{\mathrm{L}}_m}\frac{1}{2}\left(\boldsymbol{a}^{\mathrm{L}}_m-\boldsymbol{a}^{\mathrm{L}}_{m,\,\mathrm{target}}\right)^{\mathrm{T}}\left(\boldsymbol{a}^{\mathrm{L}}_m-\boldsymbol{a}^{\mathrm{L}}_{m,\,\mathrm{target}}\right),\\ &\text{subject to }\mathsf{M}^{\mathrm{L}}\boldsymbol{a}^{\mathrm{L}}_m=\boldsymbol{c}^{\mathrm{L}}_m,\quad m=1,2,3,\end{aligned} \tag{32}$$

where the vectors $\boldsymbol{a}_m^{\mathrm{L}}$ and $\boldsymbol{a}_{m,\,\mathrm{target}}^{\mathrm{L}}$ are the optimized and target volume metric terms, respectively. Herein, the analytic metric terms are the target volume metrics. Furthermore,

$$\left(\boldsymbol{a}_m^{\mathrm{L}}\right)^{\mathrm{T}} \equiv \mathbf{1}_{\mathrm{L}}^{\mathrm{T}} \left[\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m} \right]_{\mathrm{L}}, \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_2}{\partial x_m} \right]_{\mathrm{L}}, \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_3}{\partial x_m} \right]_{\mathrm{L}} \right],$$

$$\mathsf{M}^{\mathrm{L}} \equiv \left[\left(\mathsf{Q}_{\hat{\xi}_1}^{\mathrm{L}}\right)^{\mathrm{T}}, \left(\mathsf{Q}_{\hat{\xi}_2}^{\mathrm{L}}\right)^{\mathrm{T}}, \left(\mathsf{Q}_{\hat{\xi}_3}^{\mathrm{L}}\right)^{\mathrm{T}} \right],$$

and

$$\boldsymbol{c}_m^{\mathrm{L}} \equiv \frac{\Delta_2^{\mathrm{R}_f} \Delta_3^{\mathrm{R}_f}}{4} \sum_{f=1}^{4} \left\{ \mathsf{R}_{\mathrm{L}}^{\mathrm{T}} \left(\mathsf{P}_{\mathrm{L}}^{(1D)} \mathsf{I}_{\mathrm{R}_f \mathrm{toL}}^{2} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)} \mathsf{I}_{\mathrm{R}_f \mathrm{toL}}^{3} \right) \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_1}{\partial x_m} \right]_{\mathrm{R}_f}^{\hat{\Gamma}} \mathsf{R}_{\mathrm{R}_f} \right\} \mathbf{1}_{\mathrm{R}_f},$$

with $\boldsymbol{a}_m^{\mathrm{L}}$ of size $3N_{\mathrm{L}} \times 1$, M^{L} of size $N_{\mathrm{L}} \times 3N_{\mathrm{L}}$, and $\boldsymbol{c}_m^{\mathrm{L}}$ of size $N_{\mathrm{L}} \times 1$, where N_{L} is the total number of nodes in element L. The optimal solution, in the Cartesian 2-norm, is given by (see Proposition 1 in Crean et al. [14])

$$\boldsymbol{a}_m^{\mathrm{L}} = \boldsymbol{a}_{m,\,\mathrm{target}}^{\mathrm{L}} - \left(\mathsf{M}^{\mathrm{L}}\right)^{\dagger} \left(\mathsf{M}^{\mathrm{L}} \boldsymbol{a}_{m,\,\mathrm{target}}^{\mathrm{L}} - \boldsymbol{c}_m^{\mathrm{L}} \right), \tag{33}$$

where $\left(\mathsf{M}^{\mathrm{L}}\right)^{\dagger}$ is the Moore–Penrose pseudoinverse of M^{L}. This pseudoinverse is computed using a singular value decomposition of M^{L}

$$\mathsf{M}^{\mathrm{L}} = \mathsf{U}^{\mathrm{L}} \mathsf{\Sigma}^{\mathrm{L}} \left(\mathsf{V}^{\mathrm{L}}\right)^{\mathrm{T}}, \qquad \left(\mathsf{M}^{\mathrm{L}}\right)^{\dagger} = \mathsf{V}^{\mathrm{L}} \left(\mathsf{\Sigma}^{\mathrm{L}}\right)^{\dagger} \left(\mathsf{U}^{\mathrm{L}}\right)^{\mathrm{T}}.$$

The unitary matrix U^{L} is of size $N_{\mathrm{L}} \times N_{\mathrm{L}}$, $\mathsf{\Sigma}^{\mathrm{L}}$ is a diagonal matrix of size $N_{\mathrm{L}} \times N_{\mathrm{L}}$ containing the singular values of M^{L}, and $\left(\mathsf{V}^{\mathrm{L}}\right)^{\mathrm{T}}$ is of size $N_{\mathrm{L}} \times 3N_{\mathrm{L}}$ with orthonormal rows. The optimal solution $\boldsymbol{a}_m^{\mathrm{L}}$ given by Eq. (33) satisfies the discrete GCL relations (30) if the following constraint is satisfied:

$$\mathbf{1}_{\mathrm{L}}^{\mathrm{T}} \boldsymbol{c}_m^{\mathrm{L}} = 0. \tag{34}$$

The constraint (34) is a discrete approximation to the integral of the GCL equations over the domain $\hat{\Omega}_{\mathrm{L}}$, i.e.,

$$\mathbf{1}_{\mathrm{L}}^{\mathrm{T}} \boldsymbol{c}_m^{\mathrm{L}} \approx \int_{\hat{\Omega}_{\mathrm{L}}} \sum_{l=1}^{3} \frac{\partial}{\partial \hat{\xi}_l} \left(\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right) \mathrm{d}\hat{\Omega} = \oint_{\hat{\Gamma}_{\mathrm{L}}} \sum_{l=1}^{3} \hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} n_{\hat{\xi}_l} \mathrm{d}\hat{\Gamma} = 0. \tag{35}$$

In fact, our approach is to specify the surface metric terms $\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_{\mathrm{R}_f}^{\hat{\Gamma}}$ such that $\mathbf{1}_{\mathrm{L}}^{\mathrm{T}} \boldsymbol{c}_m^{\mathrm{L}}$ is exactly equal to the surface integral term on the right-hand side of Eq. (35).

The constraint Eq. (34) arises because M^{L} has one zero singular value associated with the constant singular vector. This implies that in order for Eq. (30) to have an exact solution, $\boldsymbol{c}_m^{\mathrm{L}}$ must be orthogonal to the constant vector (see Ref. [20] for a complete discussion). The next theorem is one of the main results of this work and gives the conditions on the analytic metric terms so that the constraint Eq. (34) is satisfied.

Theorem 3 *If the analytic metric terms used to populate* $\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\right]_{\mathrm{R}_f}^{\hat{\Gamma}}$ *are at most the degree of the weakest cubature rule involved in the nonconforming interface, then the constraints Eq.* (34) *are satisfied.*

Proof The proof follows from the accuracy of the interpolation operators and the associated cubature rules interacting at the nonconforming interface.

Thus far, we have concentrated on nonconforming faces. For nonconforming elements (i.e., elements that have at least one nonconforming face), on conforming faces the surface metric terms that appear in the discrete GCL Eqs. (30) and (31) are taken as the surface metric terms of the adjoining face. The metric terms of the adjoining face are approximated using a standard approach, such as that of Vinokur and Yee [59] or Thomas and Lombard [58], and Theorem 2 of Ref. [20] guarantees that metric terms computed in this way satisfy the constraint Eq. (34).

5 Nonlinearly stable schemes: Viscous Burgers' equation

The general h/p-nonconforming machinery presented in the previous section will be applied to the compressible Navier–Stokes equations in Section 6. However, in order for the resulting discretization to have the telescoping property, and therefore nonlinear stability, special nonlinear approximations are required that lead to this property. In this section, the required Hadamard derivative formulation is exemplified using the simple viscous Burgers' equation.

The viscous Burgers' equation and its canonically split form are

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial}{\partial x_1}\left(\frac{\mathcal{U}^2}{2}\right) = \frac{\partial^2 \mathcal{U}}{\partial x_1^2} \quad ; \quad \frac{\partial \mathcal{U}}{\partial t} + \frac{1}{3}\frac{\partial}{\partial x_1}\left(\mathcal{U}^2\right) + \frac{\mathcal{U}}{3}\frac{\partial \mathcal{U}}{\partial x_1} = \frac{\partial^2 \mathcal{U}}{\partial x_1^2}, \tag{36}$$

where, as in the convection-diffusion equation, the splitting is on the inviscid terms. Applying an energy analysis to the split form of Eq. (36) gives (for details, see, for example, Ref. [10])

$$\frac{1}{2}\frac{\mathrm{d}\|\mathcal{U}\|^2}{\mathrm{d}t} + \oint_\Gamma \frac{\mathcal{U}^3}{3} n_{x_1} \mathrm{d}\Gamma = \oint_\Gamma \mathcal{U}\frac{\partial \mathcal{U}}{\partial x_1} n_{x_1} \mathrm{d}\Gamma - \int_\Omega \left(\frac{\partial \mathcal{U}}{\partial x_1}\right)^2 \mathrm{d}\Omega, \ \|\mathcal{U}\|^2 \equiv \int_\Omega \mathcal{U}^2 \mathrm{d}\Omega. \tag{37}$$

The semidiscrete proof of stability that will be constructed shortly follows the continuous proof in a discrete sense such that, when contracted by $\boldsymbol{u}^{\mathrm{T}}\mathsf{P}$, i.e., the discrete analogue of multiplying by the solution and integrating in space, the result is the sum of spatial terms that telescope to the boundaries.

Ignoring the imposition of boundary conditions (i.e., SATs) and concentrating on a single element, then the discretization of Eq. (36) with SBP operators is given as

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + \frac{1}{3}\mathsf{D}_{x_1}\,\mathrm{diag}\,(\boldsymbol{u})\,\boldsymbol{u} + \frac{1}{3}\,\mathrm{diag}\,(\boldsymbol{u})\,\mathsf{D}_{x_1}\boldsymbol{u} = \mathsf{D}_{x_1}\boldsymbol{\Theta}, \quad \boldsymbol{\Theta} \equiv \mathsf{D}_{x_1}\boldsymbol{u}. \tag{38}$$

Multiplying Eq. (38) by $\boldsymbol{u}^{\mathrm{T}}\mathsf{P}$ results in

$$\frac{1}{2}\frac{\mathrm{d}\boldsymbol{u}^{\mathrm{T}}\mathsf{P}\boldsymbol{u}}{\mathrm{d}t} + \frac{1}{3}\left(\boldsymbol{u}^3(N) - \boldsymbol{u}^3(1)\right) = \boldsymbol{u}^{\mathrm{T}}\mathsf{E}_{x_1}\mathsf{D}_{x_1}\boldsymbol{u} - \boldsymbol{u}^{\mathrm{T}}\mathsf{D}_{x_1}^{\mathrm{T}}\mathsf{P}\mathsf{D}_{x_1}\boldsymbol{u}, \tag{39}$$

where each term mimics the corresponding term in Eq. (37). Furthermore, Eq. (39) has the telescoping property, i.e., the remaining terms are at the boundaries.

Notice that the key to obtaining a telescoping semidiscrete form is the careful discretization of the inviscid terms (in this case using a canonical split form), whereas the viscous terms were directly discretized in strong conservation form.

The discrete inviscid terms in Eq. (38) can be recast using the Hadamard derivative formalism. The equivalence between the split form and he Hadamard derivative operators is given as follows

$$2\mathsf{D}_{x_1} \circ \mathsf{F}_{x_1}\left(\boldsymbol{u},\boldsymbol{u}\right)\mathbf{1}_1 \quad \leftrightarrow \quad \frac{1}{3}\mathsf{D}_{x_1}\,\mathrm{diag}\left(\boldsymbol{u}\right)\boldsymbol{u} + \frac{1}{3}\,\mathrm{diag}\left(\boldsymbol{u}\right)\mathsf{D}_{x_1}\boldsymbol{u}\,. \tag{40}$$

Two components are used to construct the Hadamard derivative operator: first, an SBP derivative operator, and second, a two-point flux function related to the inviscid flux vector being discretely differentiated. The Hadamard derivative operator combines these two components such that two-point fluxes are constructed between the center point and all other points of dependency within the SBP stencil. The SBP telescoping property [26] results from precise local cancellation of spatial terms and can then be extended directly to nonlinear operators.

In the case of the Burgers' equation, the two-point flux function that results in an equivalence between the split form and the Hadamard derivative operator is [57, 10]

$$\boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{u}^{(i)},\boldsymbol{u}^{(j)}\right) \equiv \frac{\left\{\left(\boldsymbol{u}^{(i)}\right)^2 + \boldsymbol{u}^{(i)}\boldsymbol{u}^{(j)} + \left(\boldsymbol{u}^{(j)}\right)^2\right\}}{6},$$

where $\boldsymbol{u}^{(i)}$ and $\boldsymbol{u}^{(j)}$ are the i^{th} and j^{th} components of $\boldsymbol{u}$. For the purpose of demonstration, a simple SBP operator constructed on the LGL nodes $(-1, 0, 1)$ is used:

$$\mathsf{D}_{x_1} = \begin{bmatrix} -\frac{3}{2} & 2 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & -2 & \frac{3}{2} \end{bmatrix}.$$

The two argument Hadamard matrix flux, $\mathsf{F}_{x_m}\left(\boldsymbol{u},\boldsymbol{u}\right)$ is given as

$$\mathsf{F}_{x_m}\left(\boldsymbol{u},\boldsymbol{u}\right) = \begin{bmatrix} \frac{\left(\boldsymbol{u}^{(1)}\right)^2}{2} & \frac{\left(\boldsymbol{u}^{(1)}\right)^2+\boldsymbol{u}^{(1)}\boldsymbol{u}^{(2)}+\left(\boldsymbol{u}^{(2)}\right)^2}{6} & \frac{\left(\boldsymbol{u}^{(1)}\right)^2+\boldsymbol{u}^{(1)}\boldsymbol{u}^{(3)}+\left(\boldsymbol{u}^{(3)}\right)^2}{6} \\ \frac{\left(\boldsymbol{u}^{(2)}\right)^2+\boldsymbol{u}^{(2)}\boldsymbol{u}^{(1)}+\left(\boldsymbol{u}^{(1)}\right)^2}{6} & \frac{\left(\boldsymbol{u}^{(2)}\right)^2}{2} & \frac{\left(\boldsymbol{u}^{(2)}\right)^2+\boldsymbol{u}^{(2)}\boldsymbol{u}^{(3)}+\left(\boldsymbol{u}^{(3)}\right)^2}{6} \\ \frac{\left(\boldsymbol{u}^{(3)}\right)^2+\boldsymbol{u}^{(3)}\boldsymbol{u}^{(1)}+\left(\boldsymbol{u}^{(1)}\right)^2}{6} & \frac{\left(\boldsymbol{u}^{(3)}\right)^2+\boldsymbol{u}^{(3)}\boldsymbol{u}^{(2)}+\left(\boldsymbol{u}^{(2)}\right)^2}{6} & \frac{\left(\boldsymbol{u}^{(3)}\right)^2}{2} \end{bmatrix}.$$

Thus,

$$\mathsf{D}_{x_1} \circ \mathsf{F}_{x_m}\left(\boldsymbol{u},\boldsymbol{u}\right)\mathbf{1} = \begin{bmatrix} -\frac{3}{2}\frac{\left(\boldsymbol{u}^{(1)}\right)^2}{2} & 2\frac{\left(\boldsymbol{u}^{(1)}\right)^2+\boldsymbol{u}^{(1)}\boldsymbol{u}^{(2)}+\left(\boldsymbol{u}^{(2)}\right)^2}{6} & -\frac{1}{2}\frac{\left(\boldsymbol{u}^{(1)}\right)^2+\boldsymbol{u}^{(1)}\boldsymbol{u}^{(3)}+\left(\boldsymbol{u}^{(3)}\right)^2}{6} \\ -\frac{1}{2}\frac{\left(\boldsymbol{u}^{(2)}\right)^2+\boldsymbol{u}^{(2)}\boldsymbol{u}^{(1)}+\left(\boldsymbol{u}^{(1)}\right)^2}{6} & 0 & \frac{1}{2}\frac{\left(\boldsymbol{u}^{(2)}\right)^2+\boldsymbol{u}^{(2)}\boldsymbol{u}^{(3)}+\left(\boldsymbol{u}^{(3)}\right)^2}{6} \\ \frac{1}{2}\frac{\left(\boldsymbol{u}^{(3)}\right)^2+\boldsymbol{u}^{(3)}\boldsymbol{u}^{(1)}+\left(\boldsymbol{u}^{(1)}\right)^2}{6} & -2\frac{\left(\boldsymbol{u}^{(3)}\right)^2+\boldsymbol{u}^{(3)}\boldsymbol{u}^{(2)}+\left(\boldsymbol{u}^{(2)}\right)^2}{6} & \frac{3}{2}\frac{\left(\boldsymbol{u}^{(3)}\right)^2}{2} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

The equivalence between the two approaches can be determined via inspection.

The general notation necessary for discretizing the inviscid fluxes of the compressible Navier–Stokes equations is now detailed. Consider the discretization of the derivative of a flux vector $\boldsymbol{\mathcal{F}}_{x_m}$ in the x_m Cartesian direction. As for the Burgers' equation, the key components are an SBP matrix difference operator, D_{x_m}, and a two argument matrix flux function, $\mathsf{F}_{x_m}(\boldsymbol{u}_\kappa, \boldsymbol{u}_r)$, which is constructed from diagonal matrices and is defined blockwise as

$$\left(\mathsf{F}_{x_m}(\boldsymbol{u}_\kappa, \boldsymbol{u}_r)\right)(e(i-1)+1:ei, e(j-1)+1:ej) \equiv \operatorname{diag}\left(\boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{u}^{(i)}_\kappa, \boldsymbol{u}^{(j)}_r\right)\right),$$

$$\boldsymbol{u}^{(i)}_\kappa \equiv \boldsymbol{u}_\kappa(e(i-1)+1:ei), \ \boldsymbol{u}^{(j)}_r \equiv \boldsymbol{u}_r(e(j-1)+1:ej),$$

$$i = 1\ldots, N^3_\kappa, \ j = 1, \ldots, N^3_r,$$

where e is the number of equations in the system of PDEs. In the context of the compressible Navier–Stokes equations, $e = 5$ and the two argument matrix flux function is of size $\left(e\,N^3_\kappa\right) \times \left(e\,N^3_r\right)$, where eN^3_κ and eN^3_r are the total number of entries in the vectors $\boldsymbol{u}_\kappa$ and $\boldsymbol{u}_r$ corresponding the solution variables in elements κ and r, respectively. Therefore, $\boldsymbol{u}^{(i)}_\kappa$ is the vector of the e solution variables evaluated at the i^{th} node. The vectors $\boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{u}^{(i)}_\kappa, \boldsymbol{u}^{(j)}_r\right)$ are constructed from two-point flux functions that are symmetric in their arguments, $\left(\boldsymbol{u}^{(i)}_\kappa, \boldsymbol{u}^{(j)}_r\right)$, and consistent, i.e.,

$$\boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{u}^{(i)}_\kappa, \boldsymbol{u}^{(j)}_r\right) = \boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{u}^{(j)}_r, \boldsymbol{u}^{(i)}_\kappa\right), \quad \boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{u}^{(i)}_\kappa, \boldsymbol{u}^{(i)}_\kappa\right) = \boldsymbol{\mathcal{F}}_{x_m}\left(\boldsymbol{u}^{(i)}_\kappa\right),$$

where $\boldsymbol{\mathcal{F}}_{x_m}$ is the inviscid flux vector in the x_m Cartesian direction. With the notation defined, the Hadamard differentiation operator for the inviscid flux, $\frac{\partial \boldsymbol{\mathcal{F}}_{x_m}}{\partial x_m}$, is constructed as

$$2\mathsf{D}_{x_m} \circ \mathsf{F}_{x_m}(\boldsymbol{q}_\kappa, \boldsymbol{q}_\kappa)\,\mathbf{1}_\kappa \approx \frac{\partial \boldsymbol{\mathcal{F}}_{x_m}}{\partial x_m}(\boldsymbol{x}^\kappa),$$

where $\boldsymbol{x}^\kappa$ is the vector of vectors containing the nodal coordinates. The resulting approximation has equivalent order properties as constructing an approximation to the derivative of the flux vector directly using an SBP operator D_{x_m} (see Theorem 1 in Crean et al. [14]).

6 Application to the compressible Navier–Stokes equations

Herein, the nonconforming algorithm presented in Section 4 is combined with the mechanics presented in Section 5 to construct an entropy conservative discretization of the compressible Navier–Stokes equations for arbitrary h/p-nonconforming meshes. First, the continuous equations and entropy analysis are reviewed in Section 6.1. Second, in Section 6.2, the semidiscrete algorithm is presented and analyzed.

6.1 Review of the continuous entropy analysis

The entropy stable algorithm is constructed by discretizing the skew-symmetric form of the compressible Navier–Stokes equations, with the viscous flux recast in terms of entropy variables. This form of the equations is given as

$$\begin{aligned} &\mathcal{J}_\kappa \frac{\partial \boldsymbol{\mathcal{Q}}_{\boldsymbol{\kappa}}}{\partial t} + \frac{1}{2}\sum_{l,m=1}^{3}\left(\frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\boldsymbol{\mathcal{F}}^I_{x_m}\right) + \mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\frac{\partial \boldsymbol{\mathcal{F}}^I_{x_m}}{\partial \xi_l}\right) \\ &- \frac{1}{2}\sum_{l,m=1}^{3}\boldsymbol{\mathcal{F}}^I_{x_m}\frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\right) = \sum_{l,a=1}^{3}\frac{\partial}{\partial \xi_l}\left(\hat{\mathsf{C}}_{l,a}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_a}\right), \end{aligned} \tag{41}$$

where the last set of terms on the left-hand side are zero by the GCL relations Eq. (6). Furthermore,

$$\hat{\mathsf{C}}_{l,a} = \mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\sum_{m,j=1}^{3}\mathsf{C}_{m,j}\frac{\partial \xi_a}{\partial x_j}, \tag{42}$$

where $\boldsymbol{\mathcal{Q}}$ is the vector of conserved variables, and $\boldsymbol{\mathcal{F}}^I_{x_m}$ is the inviscid flux vector in the x_m direction. The vector of conserved variables is given by

$$\boldsymbol{\mathcal{Q}} = \left[\rho, \rho\mathcal{U}_1, \rho\mathcal{U}_2, \rho\mathcal{U}_3, \rho\mathcal{E}\right]^{\mathrm{T}},$$

where ρ denotes the density, $\boldsymbol{\mathcal{U}} = [\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3]^{\mathrm{T}}$ is the velocity vector, and $\mathcal{E}$ is the specific total energy. The inviscid fluxes are given as

$$\begin{aligned} \boldsymbol{\mathcal{F}}^I_{x_m} = &[\rho\mathcal{U}_m, \rho\mathcal{U}_m\mathcal{U}_1 + \delta_{m,1}\mathcal{P}, \rho\mathcal{U}_m\mathcal{U}_2 + \delta_{m,2}\mathcal{P}, \\ &\rho\mathcal{U}_m\mathcal{U}_3 + \delta_{m,3}\mathcal{P}, \rho\mathcal{U}_m\mathcal{H}]^{\mathrm{T}}, \end{aligned}$$

where $\mathcal{P}$ is the pressure, $\mathcal{H}$ is the specific total enthalpy and $\delta_{i,j}$ is the Kronecker delta.

The necessary constituent relations are

$$\mathcal{H} = c_{\mathcal{P}}\mathcal{T} + \frac{1}{2}\boldsymbol{\mathcal{U}}^{\mathrm{T}}\boldsymbol{\mathcal{U}}, \quad \mathcal{P} = \rho R\mathcal{T}, \quad R = \frac{R_u}{M_w},$$

where $\mathcal{T}$ is the temperature, R_u is the universal gas constant, M_w is the molecular weight of the gas, and $c_{\mathcal{P}}$ is the specific heat capacity at constant pressure. Finally, the specific thermodynamic entropy is given as

$$s = \frac{R}{\gamma - 1}\log\left(\frac{\mathcal{T}}{\mathcal{T}_\infty}\right) - R\log\left(\frac{\rho}{\rho_\infty}\right), \quad \gamma = \frac{c_p}{c_p - R},$$

where $\mathcal{T}_\infty$ and ρ_∞ are the reference temperature and density, respectively.

The viscous fluxes, $\boldsymbol{\mathcal{F}}^V_{x_m}$, have been recast in terms of the entropy variables, $\boldsymbol{\mathcal{W}} \equiv \partial\mathcal{S}/\partial\boldsymbol{\mathcal{Q}}$, where $\mathcal{S}$ is the entropy function $\mathcal{S} \equiv -\rho s$:

$$\boldsymbol{\mathcal{F}}^V_{x_m} = \sum_{j=1}^{3}\mathsf{C}_{m,j}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial x_j}. \tag{43}$$

The viscous fluxes written in components are given as

$$\boldsymbol{\mathcal{F}}^{V}_{x_m} = \left[0, \tau_{1,m}, \tau_{2,m}, \tau_{3,m}, \sum_{i=1}^{3} \tau_{i,m}\mathcal{U}_i - \kappa\frac{\partial \mathcal{T}}{\partial x_m}\right]^{\mathrm{T}}, \tag{44}$$

and the viscous stresses are defined as

$$\tau_{i,j} = \mu\left(\frac{\partial \mathcal{U}_i}{\partial x_j} + \frac{\partial \mathcal{U}_j}{\partial x_i} - \delta_{i,j}\frac{2}{3}\sum_{n=1}^{3}\frac{\partial \mathcal{U}_n}{\partial x_n}\right), \tag{45}$$

where $\mu(T)$ is the dynamic viscosity and $\kappa(T)$ is the thermal conductivity (not to be confused with the choice of parameter for element numbering).

The compressible Navier–Stokes equations have a convex extension, that when integrated over the physical domain, Ω, depends only on the boundary data and negative semidefinite dissipation terms. This convex extension depends on an entropy function, $\mathcal{S}$, and it is used to prove the stability in the L^2 norm. Here, a brief review of the entropy stability analysis is given. A detailed presentation is available, for instance, in Refs. [15,52,10].

Under the assumption that the entropy function $\mathcal{S}$ is convex, which is guaranteed if $\rho, \mathcal{T} > 0$, then the vector of entropy variables, $\boldsymbol{\mathcal{W}}$, simultaneously contracts all of the inviscid flux as follows:

$$\boldsymbol{\mathcal{W}}^{\mathrm{T}}\frac{\partial \boldsymbol{\mathcal{F}}^{I}_{x_m}}{\partial \xi_l} = \frac{\partial \mathcal{S}}{\partial \boldsymbol{\mathcal{Q}}}\frac{\partial \boldsymbol{\mathcal{F}}^{I}_{x_m}}{\partial \boldsymbol{\mathcal{Q}}}\frac{\partial \boldsymbol{\mathcal{Q}}}{\partial \xi_l} = \frac{\partial \mathcal{F}_{x_m}}{\partial \boldsymbol{\mathcal{Q}}}\frac{\partial \boldsymbol{\mathcal{F}}^{I}_{x_m}}{\partial \boldsymbol{\mathcal{Q}}}, \quad l,\, m = 1,\, 2,\, 3, \tag{46}$$

where $\mathcal{F}_{x_m}$ is the entropy flux in the x_m direction.

The entropy stability analysis proceeds by first multiplying (contracting) Eq. (41) by the transpose of the entropy variables, $\boldsymbol{\mathcal{W}}^{\mathrm{T}}$,

$$\overbrace{\mathcal{J}_\kappa \boldsymbol{\mathcal{W}}^{\mathrm{T}}\frac{\partial \boldsymbol{\mathcal{Q}}_{\boldsymbol{\kappa}}}{\partial t}}^{I} + \frac{1}{2}\sum_{l,m=1}^{3}\left(\overbrace{\boldsymbol{\mathcal{W}}^{\mathrm{T}}\frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\boldsymbol{\mathcal{F}}^{I}_{x_m}\right)}^{II} + \overbrace{\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\boldsymbol{\mathcal{W}}^{\mathrm{T}}\frac{\partial \boldsymbol{\mathcal{F}}^{I}_{x_m}}{\partial \xi_l}}^{III}\right) = \sum_{l,a=1}^{3}\overbrace{\boldsymbol{\mathcal{W}}^{\mathrm{T}}\frac{\partial}{\partial \xi_l}\left(\hat{\mathsf{C}}_{l,a}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_a}\right)}^{IV}. \tag{47}$$

With the help of Eq. (46) and the product rule, the terms $I - IV$ are now simplified:

$$I \equiv \mathcal{J}_\kappa \boldsymbol{\mathcal{W}}^{\mathrm{T}}\frac{\partial \boldsymbol{\mathcal{Q}}_{\boldsymbol{\kappa}}}{\partial t} = \mathcal{J}_\kappa \frac{\partial \mathcal{S}}{\partial \boldsymbol{\mathcal{Q}}}\frac{\partial \boldsymbol{\mathcal{Q}}_{\boldsymbol{\kappa}}}{\partial t} = \mathcal{J}_\kappa \frac{\partial \mathcal{S}_\kappa}{\partial t}, \tag{48}$$

$$\begin{aligned} II \equiv \boldsymbol{\mathcal{W}}^{\mathrm{T}}\frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\boldsymbol{\mathcal{F}}^{I}_{x_m}\right) &= \mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\boldsymbol{\mathcal{W}}^{\mathrm{T}}\frac{\partial \boldsymbol{\mathcal{F}}^{I}_{x_m}}{\partial \xi_l} + \boldsymbol{\mathcal{W}}^{\mathrm{T}}\boldsymbol{\mathcal{F}}^{I}_{x_m}\frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\right) \\ &= \mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\frac{\partial \mathcal{F}_{x_m}}{\partial \xi_l} + \boldsymbol{\mathcal{W}}^{\mathrm{T}}\boldsymbol{\mathcal{F}}^{I}_{x_m}\frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\right), \end{aligned} \tag{49}$$

$$III \equiv \mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\boldsymbol{\mathcal{W}}^{\mathrm{T}}\frac{\partial \boldsymbol{\mathcal{F}}^{I}_{x_m}}{\partial \xi_l} = \mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\frac{\partial \mathcal{F}_{x_m}}{\partial \xi_l}, \tag{50}$$

$$IV \equiv \boldsymbol{\mathcal{W}}^{\mathrm{T}} \frac{\partial}{\partial \xi_l}\left(\hat{\mathsf{C}}_{l,a}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_a}\right) = \frac{\partial}{\partial \xi_l}\left(\boldsymbol{\mathcal{W}}^{\mathrm{T}}\hat{\mathsf{C}}_{l,a}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_a}\right) - \frac{\partial \boldsymbol{\mathcal{W}}^{\mathrm{T}}}{\partial \xi_l}\hat{\mathsf{C}}_{l,a}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_a}. \tag{51}$$

Substituting Eqs. (48) through (51) into Eq. (47) results in

$$\begin{aligned}\mathcal{J}_\kappa \frac{\partial \mathcal{S}_\kappa}{\partial t} + \sum_{l,m=1}^{3} \mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\frac{\partial \mathcal{F}_{x_m}}{\partial \xi_l} + \frac{\boldsymbol{\mathcal{W}}^{\mathrm{T}}}{2}\sum_{m=1}^{3}\boldsymbol{\mathcal{F}}^I_{x_m}\cancelto{0 \text{ via GCL}(6)}{\sum_{l=1}^{3}\frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\right)} = \\ \sum_{l,a=1}^{3}\left\{\frac{\partial}{\partial \xi_l}\left(\boldsymbol{\mathcal{W}}^{\mathrm{T}}\hat{\mathsf{C}}_{l,a}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_a}\right) - \frac{\partial \boldsymbol{\mathcal{W}}^{\mathrm{T}}}{\partial \xi_l}\hat{\mathsf{C}}_{l,a}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_a}\right\}.\end{aligned} \tag{52}$$

Bringing the metric terms within the derivative on the term $\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\frac{\partial \mathcal{F}_{x_m}}{\partial \xi_l}$ and using the product rule results in

$$\begin{aligned}\mathcal{J}_\kappa \frac{\partial \mathcal{S}_\kappa}{\partial t} + \sum_{l,m=1}^{3}\frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\mathcal{F}_{x_m}\right) - \sum_{m=1}^{3}\mathcal{F}_{x_m}\cancelto{0 \text{ via GCL } (6)}{\sum_{m=1}^{3}\frac{\partial}{\partial \xi_l}\left(\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\right)} = \\ \sum_{l,a=1}^{3}\left\{\frac{\partial}{\partial \xi_l}\left(\boldsymbol{\mathcal{W}}^{\mathrm{T}}\hat{\mathsf{C}}_{l,a}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_a}\right) - \frac{\partial \boldsymbol{\mathcal{W}}^{\mathrm{T}}}{\partial \xi_l}\hat{\mathsf{C}}_{l,a}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_a}\right\}.\end{aligned} \tag{53}$$

Rearranging Eq. (53) and expanding the dissipation term yields

$$\begin{aligned}\mathcal{J}_\kappa \frac{\partial \mathcal{S}_\kappa}{\partial t} = \sum_{l=1}^{3}\frac{\partial}{\partial \xi_l}\left(-\sum_{m=1}^{3}\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\mathcal{F}_{x_m} + \sum_{a=1}^{3}\left(\boldsymbol{\mathcal{W}}^{\mathrm{T}}\hat{\mathsf{C}}_{l,a}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_a}\right)\right) \\ - \begin{bmatrix}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_1}\\ \frac{\partial \hat{\boldsymbol{\mathcal{W}}}}{\partial \xi_2}\\ \frac{\partial \hat{\boldsymbol{\mathcal{W}}}}{\partial \xi_3}\end{bmatrix}^{\mathrm{T}} \underbrace{\begin{bmatrix}\hat{\mathsf{C}}_{1,1} & \hat{\mathsf{C}}_{1,2} & \hat{\mathsf{C}}_{1,3}\\ \hat{\mathsf{C}}_{1,2}^{\mathrm{T}} & \hat{\mathsf{C}}_{2,2} & \hat{\mathsf{C}}_{2,3}\\ \hat{\mathsf{C}}_{1,3}^{\mathrm{T}} & \hat{\mathsf{C}}_{2,3}^{\mathrm{T}} & \hat{\mathsf{C}}_{3,3}\end{bmatrix}}_{\equiv \hat{\mathsf{C}}} \underbrace{\begin{bmatrix}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_1}\\ \frac{\partial \hat{\boldsymbol{\mathcal{W}}}}{\partial \xi_2}\\ \frac{\partial \hat{\boldsymbol{\mathcal{W}}}}{\partial \xi_3}\end{bmatrix}}_{\equiv \hat{\boldsymbol{\mathcal{W}}}},\end{aligned} \tag{54}$$

where the matrix $\hat{\mathsf{C}}$ is symmetric semidefinite (see Ref. [24] for details).

Integrating Eq. (54) in space and using integration by parts gives

$$\int_{\hat{\Omega}_\kappa}\mathcal{J}_\kappa \frac{\partial \mathcal{S}_\kappa}{\partial t}\mathrm{d}\hat{\Omega} \leq \sum_{l=1}^{3}\oint_{\hat{\Gamma}_\kappa}\left(-\sum_{m=1}^{3}\mathcal{J}_\kappa \frac{\partial \xi_l}{\partial x_m}\mathcal{F}_{x_m} + \sum_{a=1}^{3}\left(\boldsymbol{\mathcal{W}}^{\mathrm{T}}\hat{\mathsf{C}}_{l,a}\frac{\partial \boldsymbol{\mathcal{W}}}{\partial \xi_a}\right)\right) n_{\xi_l}\mathrm{d}\hat{\Gamma}. \tag{55}$$

An L^2 bound on the solution is derived from inequality Eq. (55) by integrating in time and assuming 1) nonlinearly well-posed boundary and initial conditions, and 2) positivity of temperature and density. Then, the result can be turned into a bound on the solution in terms of the data of the problem [15,52].

6.2 An h/p-nonconforming algorithm

The skew-symmetrically split form of the compressible Navier–Stokes equations (41) is discretized by combining the macro element SBP operator in Section 4.2 with the nonlinear mechanics presented in Section 5. Thus, the discretization of Eq. (41) over the macro element is given as

$$\begin{aligned}
&\hat{\mathsf{J}}\frac{\mathrm{d}\widetilde{\boldsymbol{q}}}{\partial t} + \sum_{l,m=1}^{3} \widetilde{\mathsf{D}}_{l,m} \circ \mathsf{F}_{x_m}\left(\widetilde{\boldsymbol{q}},\widetilde{\boldsymbol{q}}\right)\tilde{\mathbf{1}} \\
&- \frac{1}{2}\sum_{l,m=1}^{3} \mathrm{diag}\left(\boldsymbol{f}^{I}_{x_m}\right)\widetilde{\mathsf{D}}_{\hat{\xi}_l}\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m}\right]\tilde{\mathbf{1}} = \sum_{l,a=1}^{3}\widetilde{\mathsf{D}}_{\hat{\xi}_l}\left[\hat{\mathsf{C}}_{l,a}\right]\widetilde{\mathsf{D}}_{\hat{\xi}_a}\widetilde{\boldsymbol{w}}, \\
&\widetilde{\boldsymbol{q}} \equiv \left[\boldsymbol{q}_{\mathrm{L}}^{\mathrm{T}}, \boldsymbol{q}_{\mathrm{R}_1}^{\mathrm{T}}, \ldots, \boldsymbol{q}_{\mathrm{R}_4}^{\mathrm{T}}\right]^{\mathrm{T}}, \ \widetilde{\boldsymbol{w}} \equiv \left[\boldsymbol{w}_{\mathrm{L}}^{\mathrm{T}}, \boldsymbol{w}_{\mathrm{R}_1}^{\mathrm{T}}, \ldots, \boldsymbol{w}_{\mathrm{R}_4}^{\mathrm{T}}\right]^{\mathrm{T}},
\end{aligned} \tag{56}$$

where $\boldsymbol{f}^{I}_{x_m}$ is a vector of vectors constructed by evaluating $\boldsymbol{\mathcal{F}}^{I}_{x_m}$ at the mesh nodes. Note that the factor of $\frac{1}{2}$ on the skew-symmetric inviscid volume terms has been absorbed as a result of using the nonlinear operator, e.g., $2\mathsf{D}_{\xi_l} \circ \mathsf{F}_{x_m}\left(\boldsymbol{q}_\kappa, \boldsymbol{q}_\kappa\right)\mathbf{1}_\kappa \approx \frac{\partial \boldsymbol{\mathcal{F}}_{x_m}}{\partial \xi_l}(\boldsymbol{\xi}^\kappa)$. Furthermore, the flux function matrix, $\mathsf{F}_{x_m}\left(\widetilde{\boldsymbol{q}},\widetilde{\boldsymbol{q}}\right)$, is constructed using a two-point flux function, $\boldsymbol{f}^{sc}_{x_m}\left(\tilde{\boldsymbol{q}}^{(i)}, \tilde{\boldsymbol{q}}^{(i)}\right)$, that satisfies the Tadmor's shuffle condition [57]

$$\left(\tilde{\boldsymbol{w}}^{(i)} - \tilde{\boldsymbol{w}}^{(j)}\right)^{\mathrm{T}} \boldsymbol{f}^{sc}_{x_m}\left(\tilde{\boldsymbol{q}}^{(i)}, \tilde{\boldsymbol{q}}^{(i)}\right) = \tilde{\boldsymbol{\psi}}^{(i)}_{x_m} - \tilde{\boldsymbol{\psi}}^{(j)}_{x_m}. \tag{57}$$

The $\widetilde{\mathsf{D}}_{l,m}$ operators are constructed from the scalar conservation law operators developed in Section (4) by tensoring them with an identity matrix, I_5, to accommodate the system of five equations. For example,

$$\mathsf{D}^{\mathrm{L}}_{\hat{\xi}_1} \equiv \overline{\mathsf{D}}_{\hat{\xi}_1} \otimes \mathsf{I}_5, \quad \overline{\mathsf{D}}^{\mathrm{L}}_{\hat{\xi}_1} \equiv \frac{2}{\Delta^{\mathrm{L}}_1}\mathsf{D}^{(1D)}_{\mathrm{L}} \otimes \mathsf{I}_{\mathrm{L}} \otimes \mathsf{I}_{\mathrm{L}}.$$

Similar to the linear stability, entropy stability necessitates that the last set of terms on the left-hand side of Eq. (56) be zero and leads to the same set of discrete GCL conditions.

The semidiscrete entropy analysis follows the continuous analysis in a one-to-one fashion. In order to simplify the derivation, the following matrices are introduced:

$$\tilde{\mathsf{D}}_m \equiv \sum_{l=1}^{3}\widetilde{\mathsf{D}}_{l,m}, \quad \tilde{\mathsf{Q}}_m \equiv \widetilde{\mathsf{P}}\widetilde{\mathsf{D}}_m, \quad \tilde{\mathsf{E}}_m \equiv \tilde{\mathsf{Q}}_m + \tilde{\mathsf{Q}}_m^{\mathrm{T}}.$$

Assuming that the discrete GCL conditions are satisfied, Eq. (56) becomes

$$\hat{\mathsf{J}}\frac{\mathrm{d}\widetilde{\boldsymbol{q}}}{\partial t} + \sum_{m=1}^{3}\tilde{\mathsf{D}}_m \circ \mathsf{F}_{x_m}\left(\widetilde{\boldsymbol{q}},\widetilde{\boldsymbol{q}}\right)\tilde{\mathbf{1}} = \sum_{l,a=1}^{3}\widetilde{\mathsf{D}}_{\hat{\xi}_l}\left[\hat{\mathsf{C}}_{l,a}\right]\widetilde{\mathsf{D}}_{\hat{\xi}_a}\widetilde{\boldsymbol{w}}. \tag{58}$$

Multiplying Eq. (58) by $\widetilde{\boldsymbol{w}}^{\mathrm{T}}\widetilde{\mathsf{P}}$ (the discrete analogue of multiplying by $\boldsymbol{\mathcal{W}}^{\mathrm{T}}$ and integrating over the domain) yields

$$\hat{\mathsf{J}}\widetilde{\boldsymbol{w}}^{\mathrm{T}}\widetilde{\mathsf{P}}\frac{\mathrm{d}\widetilde{\boldsymbol{q}}}{\partial t}+\sum_{m=1}^{3}\widetilde{\boldsymbol{w}}\tilde{\mathsf{Q}}_m\circ\mathsf{F}_{x_m}\left(\widetilde{\boldsymbol{q}},\widetilde{\boldsymbol{q}}\right)\tilde{\mathbf{1}}=\sum_{l,a=1}^{3}\widetilde{\boldsymbol{w}}^{\mathrm{T}}\widetilde{\mathsf{Q}}_{\hat{\xi}_l}\left[\hat{\mathsf{C}}_{l,a}\right]\widetilde{\mathsf{D}}_{\hat{\xi}_a}\widetilde{\boldsymbol{w}}. \tag{59}$$

Taking the transpose of one half of the volume term on the left-hand side of Eq. (59), using the SBP property $\mathsf{Q}=-\mathsf{Q}^{\mathrm{T}}+\mathsf{E}$, and the symmetry of the two-point flux function matrix, results in

$$\begin{aligned}\hat{\mathsf{J}}\widetilde{\boldsymbol{w}}^{\mathrm{T}}\widetilde{\mathsf{P}}\frac{\mathrm{d}\widetilde{\boldsymbol{q}}}{\partial t}+\frac{1}{2}\sum_{m=1}^{3}\Big(&\widetilde{\boldsymbol{w}}\tilde{\mathsf{Q}}_m\circ\mathsf{F}_{x_m}\left(\widetilde{\boldsymbol{q}},\widetilde{\boldsymbol{q}}\right)\tilde{\mathbf{1}}-\tilde{\mathbf{1}}^{\mathrm{T}}\tilde{\mathsf{Q}}_m\circ\mathsf{F}_{x_m}\left(\widetilde{\boldsymbol{q}},\widetilde{\boldsymbol{q}}\right)^{\mathrm{T}}\widetilde{\boldsymbol{w}}\\&+\tilde{\mathbf{1}}^{\mathrm{T}}\tilde{\mathsf{E}}_m\circ\mathsf{F}_{x_m}\left(\widetilde{\boldsymbol{q}},\widetilde{\boldsymbol{q}}\right)^{\mathrm{T}}\widetilde{\boldsymbol{w}}\Big)=\\&\sum_{l,a=1}^{3}\widetilde{\boldsymbol{w}}^{\mathrm{T}}\widetilde{\mathsf{E}}_{\hat{\xi}_l}\left[\hat{\mathsf{C}}_{l,a}\right]\widetilde{\mathsf{D}}_{\hat{\xi}_a}\widetilde{\boldsymbol{w}}-\sum_{l,a=1}^{3}\widetilde{\boldsymbol{w}}^{\mathrm{T}}\widetilde{\mathsf{D}}_{\hat{\xi}_l}^{\mathrm{T}}\widetilde{\mathsf{P}}\left[\hat{\mathsf{C}}_{l,a}\right]\widetilde{\mathsf{D}}_{\hat{\xi}_a}\widetilde{\boldsymbol{w}}.\end{aligned} \tag{60}$$

To further reduce the left-hand side terms requires the following theorem (this is Theorem 8 in Ref. [18] and the proof is given in Appendix D of that document):

Theorem 4 *Consider the matrix of* $\overline{\mathsf{A}}$ *of size* $N_\kappa\times N_r$ *with a tensor extension* $\mathsf{A}\equiv\overline{\mathsf{A}}\otimes\mathsf{I}_5$*, and a two argument matrix flux function* $\mathsf{F}_{x_m}\left(\boldsymbol{q}_\kappa,\boldsymbol{q}_r\right)$ *constructed from the two-point flux function* $\boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{q}_\kappa^{(i)},\boldsymbol{q}_r^{(j)}\right)$ *that satisfies the Tadmor's shuffle condition*

$$\left(\boldsymbol{w}_\kappa^{(i)}-\boldsymbol{w}_r^{(j)}\right)^{\mathrm{T}}\boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{q}_\kappa^{(i)},\boldsymbol{q}_r^{(j)}\right)=\left(\boldsymbol{\psi}^{\kappa}_{x_m}\right)^{(i)}-\left(\boldsymbol{\psi}^{r}_{x_m}\right)^{(j)}$$

and is symmetric, i.e., $\boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{q}_\kappa^{(i)},\boldsymbol{q}_r^{(j)}\right)=\boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{q}_r^{(j)},\boldsymbol{q}_\kappa^{(i)}\right)$*, then*

$$\boldsymbol{w}_\kappa^{\mathrm{T}}\left(\mathsf{A}\right)\circ\mathsf{F}_{x_m}\left(\boldsymbol{q}_\kappa,\boldsymbol{q}_r\right)\mathbf{1}_r-\mathbf{1}_\kappa^{\mathrm{T}}\mathsf{A}\circ\mathsf{F}_{x_m}\left(\boldsymbol{q}_\kappa,\boldsymbol{q}_r\right)\boldsymbol{w}_r=\left(\boldsymbol{\psi}^{\kappa}_{x_m}\right)^{\mathrm{T}}\overline{\mathsf{A}}\overline{\mathbf{1}}_r-\overline{\mathbf{1}}_\kappa^{\mathrm{T}}\overline{\mathsf{A}}\boldsymbol{\psi}^{r}_{x_m}.$$

Proof The proof is given in Ref. [18] and we reproduce it here for completeness.

$$\begin{aligned}&\boldsymbol{w}_\kappa^{\mathrm{T}}\left(\mathsf{A}\right)\circ\mathsf{F}_{x_m}\left(\boldsymbol{q}_\kappa,\boldsymbol{q}_r\right)\mathbf{1}_r-\mathbf{1}_\kappa^{\mathrm{T}}\mathsf{A}\circ\mathsf{F}_{x_m}\left(\boldsymbol{q}_\kappa,\boldsymbol{q}_r\right)\boldsymbol{w}_r=\\&\sum_{i=1}^{N_\kappa^3}\sum_{j=1}^{N_r^3}\left\{\overline{\mathsf{A}}(i,j)\left(\boldsymbol{w}_\kappa^{(i)}\right)^{\mathrm{T}}\boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{q}_\kappa^{(i)},\boldsymbol{q}_r^{(j)}\right)-\overline{\mathsf{A}}(i,j)\left(\boldsymbol{w}_r^{(j)}\right)^{\mathrm{T}}\boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{q}_\kappa^{(i)},\boldsymbol{q}_r^{(j)}\right)\right\}=\\&\sum_{i=1}^{N_\kappa^3}\sum_{j=1}^{N_r^3}\left\{\overline{\mathsf{A}}(i,j)\left(\left(\boldsymbol{w}_\kappa^{(i)}\right)^{\mathrm{T}}-\left(\boldsymbol{w}_r^{(j)}\right)^{\mathrm{T}}\right)\boldsymbol{f}^{sc}_{x_m}\left(\boldsymbol{q}_\kappa^{(i)},\boldsymbol{q}_r^{(j)}\right)\right\}=\\&\sum_{i=1}^{N_\kappa^3}\sum_{j=1}^{N_r^3}\overline{\mathsf{A}}(i,j)\left(\left(\boldsymbol{\psi}^{\kappa}_{x_m}\right)^{(i)}-\left(\boldsymbol{\psi}^{r}_{x_m}\right)^{(j)}\right)=\\&\left(\boldsymbol{\psi}^{\kappa}_{x_m}\right)^{\mathrm{T}}\overline{\mathsf{A}}\overline{\mathbf{1}}_r-\overline{\mathbf{1}}_\kappa^{\mathrm{T}}\overline{\mathsf{A}}\boldsymbol{\psi}^{r}_{x_m}.\end{aligned}$$

Applying Theorem 4 to the volume terms on the left-hand side of Eq. (60) yields

$$\hat{\mathsf{J}}\widetilde{\boldsymbol{w}}^{\mathrm{T}}\widetilde{\mathsf{P}}\frac{\mathrm{d}\widetilde{\boldsymbol{q}}}{\partial t}+\frac{1}{2}\sum_{m=1}^{3}\left\{\left(\widetilde{\psi_{x_m}}\right)^{\mathrm{T}}\overline{\widetilde{\mathsf{Q}}}_m\overline{\mathbf{1}}-\overline{\mathbf{1}}^{\mathrm{T}}\overline{\widetilde{\mathsf{Q}}}_m\widetilde{\psi_{x_m}}+\tilde{\mathbf{1}}^{\mathrm{T}}\tilde{\mathsf{E}}_m\circ\mathsf{F}_{x_m}\left(\widetilde{\boldsymbol{q}},\widetilde{\boldsymbol{q}}\right)^{\mathrm{T}}\widetilde{\boldsymbol{w}}\right\}= \sum_{l,a=1}^{3}\widetilde{\boldsymbol{w}}^{\mathrm{T}}\widetilde{\mathsf{E}}_{\hat{\xi}_l}\left[\hat{\mathsf{C}}_{l,a}\right]\widetilde{\mathsf{D}}_{\hat{\xi}_a}\widetilde{\boldsymbol{w}}-\sum_{l,a=1}^{3}\widetilde{\boldsymbol{w}}^{\mathrm{T}}\widetilde{\mathsf{D}}_{\hat{\xi}_l}^{\mathrm{T}}\widetilde{\mathsf{P}}\left[\hat{\mathsf{C}}_{l,a}\right]\widetilde{\mathsf{D}}_{\hat{\xi}_a}\widetilde{\boldsymbol{w}}. \tag{61}$$

The term $\overline{\widetilde{\mathsf{Q}}}_m\overline{\mathbf{1}}$ is zero by the discrete GCL conditions and the consistency of the derivative operator ($\mathsf{D}\mathbf{1}=\mathbf{0}$) and for the same reasons $\overline{\mathbf{1}}^{\mathrm{T}}\overline{\widetilde{\mathsf{Q}}}_m=\overline{\mathbf{1}}^{\mathrm{T}}\overline{\widetilde{\mathsf{E}}}_m$. Therefore, after some rearrangements Eq. (62) reduces to

$$\hat{\mathsf{J}}\widetilde{\boldsymbol{w}}^{\mathrm{T}}\widetilde{\mathsf{P}}\frac{\mathrm{d}\widetilde{\boldsymbol{q}}}{\partial t}=-\frac{1}{2}\sum_{m=1}^{3}\left\{-\overline{\mathbf{1}}^{\mathrm{T}}\overline{\widetilde{\mathsf{E}}}_m\widetilde{\psi_{x_m}}+\tilde{\mathbf{1}}^{\mathrm{T}}\tilde{\mathsf{E}}_m\circ\mathsf{F}_{x_m}\left(\widetilde{\boldsymbol{q}},\widetilde{\boldsymbol{q}}\right)^{\mathrm{T}}\widetilde{\boldsymbol{w}}\right\}+\sum_{l,a=1}^{3}\widetilde{\boldsymbol{w}}^{\mathrm{T}}\widetilde{\mathsf{E}}_{\hat{\xi}_l}\left[\hat{\mathsf{C}}_{l,a}\right]\widetilde{\mathsf{D}}_{\hat{\xi}_a}\widetilde{\boldsymbol{w}}-\sum_{l,a=1}^{3}\widetilde{\boldsymbol{w}}^{\mathrm{T}}\widetilde{\mathsf{D}}_{\hat{\xi}_l}^{\mathrm{T}}\widetilde{\mathsf{P}}\left[\hat{\mathsf{C}}_{l,a}\right]\widetilde{\mathsf{D}}_{\hat{\xi}_a}\widetilde{\boldsymbol{w}}. \tag{62}$$

The right-hand side of Eq. (62) contains surface terms (those constructed from the E matrices) and viscous dissipation terms (the last set of terms). The surface terms can be decomposed into the contributions of the separate surfaces of the element (nodewise). The entropy conservation of the algorithm follows immediately for periodic problems because these terms would cancel out with the contributions from the coupling SATs. For general boundary conditions, appropriate SATs need to be constructed so that an entropy inequality or equality is attained (see, for example, [44,53,16]).

As in the section dedicated to the convection-diffusion equation, here we present the elementwise formulation of the discretization (in Appendix B, further details are provided to help the reader in the implementation of the algorithm). On the L element, the discretization reads

$$\begin{aligned}
&\hat{\mathsf{J}}_{\mathrm{L}}\frac{\mathrm{d}\boldsymbol{u}_{\mathrm{L}}}{\mathrm{d}t}+\sum_{l,m=1}^{3}\left(\mathsf{D}_{\hat{\xi}_l}^{\mathrm{L}}\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_l}{\partial x_m}\right]_{\mathrm{L}}+\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_l}{\partial x_m}\right]_{\mathrm{L}}\mathsf{D}_{\hat{\xi}_l}^{\mathrm{L}}\right)\circ\mathsf{F}_{x_m}\left(\boldsymbol{u}_{\mathrm{L}},\boldsymbol{u}_{\mathrm{L}}\right)\mathbf{1}_{\mathrm{L}}=\\
&\sum_{l,a=1}^{3}\mathsf{D}_{\hat{\xi}_l}^{\mathrm{L}}\left[\hat{\mathsf{C}}_{l,a}\right]\boldsymbol{\theta}_a^{\mathrm{L}}\\
&+\mathsf{P}_{\mathrm{L}}^{-1}\sum_{l=1}^{3}\left\{\left(\boldsymbol{e}_N^{\mathrm{L}}\left(\boldsymbol{e}_N^{\mathrm{L}}\right)^{\mathrm{T}}\otimes\mathsf{P}_{\mathrm{L}}^{(1D)}\otimes\mathsf{P}_{\mathrm{L}}^{(1D)}\otimes\mathsf{I}_5\right)\circ\mathsf{F}_{x_m}\left(\boldsymbol{u}_{\mathrm{L}},\boldsymbol{u}_{\mathrm{L}}\right)\mathbf{1}_{\mathrm{L}}\right.\\
&\left.-\sum_{f=1}^{4}\left(\boldsymbol{e}_N^{\mathrm{L}}\left(\boldsymbol{e}_1^{\mathrm{R}_f}\right)^{\mathrm{T}}\otimes\mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^2_{\mathrm{R}_f\mathrm{toL}}\otimes\mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^3_{\mathrm{R}_f\mathrm{toL}}\otimes\mathsf{I}_5\right)\circ\mathsf{F}_{x_m}\left(\boldsymbol{u}_{\mathrm{L}},\boldsymbol{u}_{\mathrm{R}_f}\right)\mathbf{1}_{\mathrm{R}_f}\right\}\\
&-\frac{1}{2}\mathsf{P}_{\mathrm{L}}^{-1}\sum_{l=1}^{3}\left\{\left(\boldsymbol{e}_N^{\mathrm{L}}\left(\boldsymbol{e}_N^{\mathrm{L}}\right)^{\mathrm{T}}\otimes\mathsf{P}_{\mathrm{L}}^{(1D)}\otimes\mathsf{P}_{\mathrm{L}}^{(1D)}\otimes\mathsf{I}_5\right)\boldsymbol{\theta}_a^{\mathrm{L}}\right.\\
&\left.-\sum_{f=1}^{4}\left(\boldsymbol{e}_N^{\mathrm{L}}\left(\boldsymbol{e}_1^{\mathrm{R}_f}\right)^{\mathrm{T}}\otimes\mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^2_{\mathrm{R}_f\mathrm{toL}}\otimes\mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^3_{\mathrm{R}_f\mathrm{toL}}\otimes\mathsf{I}_5\right)\boldsymbol{\theta}_a^{\mathrm{R}_f}\right\}+\boldsymbol{SAT}_{\mathrm{L}}^1,
\end{aligned} \tag{63}$$

where,

$$\begin{aligned}\boldsymbol{\theta}_a^{\mathrm{L}} &= \mathsf{D}_{\hat{\xi}_a}^{\mathrm{L}} \boldsymbol{w}_{\mathrm{L}} \\ &- \frac{1}{2}\mathsf{P}_{\mathrm{L}}^{-1} \sum_{l=1}^{3} \left\{ \left(\boldsymbol{e}_N^{\mathrm{L}} \left(\boldsymbol{e}_N^{\mathrm{L}} \right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)} \otimes \mathsf{I}_5 \right) \boldsymbol{w}_{\mathrm{L}} \right. \\ &\left. - \sum_{f=1}^{4} \left(\boldsymbol{e}_N^{\mathrm{L}} \left(\boldsymbol{e}_1^{\mathrm{R}_f} \right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)} \mathsf{I}^2_{\mathrm{R}_f \mathrm{toL}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)} \mathsf{I}^3_{\mathrm{R}_f \mathrm{toL}} \otimes \mathsf{I}_5 \right) \boldsymbol{w}_{\mathrm{R}_f} \right\} + \boldsymbol{SAT}_{\mathrm{L}}^2,\end{aligned} \tag{64}$$

and $\boldsymbol{SAT}_{\mathrm{L}}^1$ and $\boldsymbol{SAT}_{\mathrm{L}}^2$ contain the SATs for the remaining five faces on element L.

Similarly, on the R_f elements

$$\begin{aligned}&\hat{\mathsf{J}}_{\mathrm{R}_f} \frac{\mathrm{d}\boldsymbol{u}_{\mathrm{R}_f}}{\mathrm{d}t} + \sum_{l,m=1}^{3} \left(\mathsf{D}_{\hat{\xi}_l}^{\mathrm{R}_f} \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_{\mathrm{R}_f} + \left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]_{\mathrm{R}_f} \mathsf{D}_{\hat{\xi}_l}^{\mathrm{R}_f} \right) \circ \mathsf{F}_{x_m} \left(\boldsymbol{u}_{\mathrm{R}_f}, \boldsymbol{u}_{\mathrm{R}_f} \right) \mathbf{1}_{\mathrm{R}_f} = \\ &\sum_{l,a=1}^{3} \mathsf{D}_{\hat{\xi}_l}^{\mathrm{R}_f} \left[\hat{\mathsf{C}}_{l,a} \right] \boldsymbol{\theta}_a^{\mathrm{R}_f} \\ &- \mathsf{P}_{\mathrm{R}_f}^{-1} \sum_{l=1}^{3} \left\{ \left(\boldsymbol{e}_1^{\mathrm{R}_f} \left(\boldsymbol{e}_1^{\mathrm{R}_f} \right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \otimes \mathsf{I}_5 \right) \circ \mathsf{F}_{x_m} \left(\boldsymbol{u}_{\mathrm{R}_f}, \boldsymbol{u}_{\mathrm{R}_f} \right) \mathbf{1}_{\mathrm{R}_f} \right. \\ &\left. - \sum_{f=1}^{4} \left(\boldsymbol{e}_1^{\mathrm{R}_f} \left(\boldsymbol{e}_N^{\mathrm{L}} \right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \mathsf{I}^2_{\mathrm{LtoR_f}} \otimes \mathsf{P}_{\mathrm{R}}^{(1D)} \mathsf{I}^3_{\mathrm{LtoR_f}} \otimes \mathsf{I}_5 \right) \circ \mathsf{F}_{x_m} \left(\boldsymbol{u}_{\mathrm{R}_f}, \boldsymbol{u}_{\mathrm{L}} \right) \mathbf{1}_{\mathrm{L}} \right\} \\ &- \frac{1}{2}\mathsf{P}_{\mathrm{R}_f}^{-1} \sum_{l=1}^{3} \left\{ \left(\boldsymbol{e}_1^{\mathrm{R}_f} \left(\boldsymbol{e}_1^{\mathrm{R}_f} \right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \otimes \mathsf{I}_5 \right) \boldsymbol{\theta}_a^{\mathrm{R}_f} \right. \\ &\left. - \sum_{f=1}^{4} \left(\boldsymbol{e}_1^{\mathrm{R}_f} \left(\boldsymbol{e}_N^{\mathrm{L}} \right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \mathsf{I}^2_{\mathrm{LtoR_f}} \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \mathsf{I}^3_{\mathrm{LtoR_f}} \otimes \mathsf{I}_5 \right) \boldsymbol{\theta}_a^{\mathrm{L}} \right\} + \boldsymbol{SAT}_{\mathrm{R}_f}^1,\end{aligned} \tag{65}$$

where,

$$\begin{aligned}\boldsymbol{\theta}_a^{\mathrm{R}_f} &= \mathsf{D}_{\hat{\xi}_a}^{\mathrm{R}_f} \boldsymbol{w}_{\mathrm{R}_f} \\ &- \frac{1}{2}\mathsf{P}_{\mathrm{R}_f}^{-1} \sum_{l=1}^{3} \left\{ \left(\boldsymbol{e}_1^{\mathrm{R}_f} \left(\boldsymbol{e}_1^{\mathrm{R}_f} \right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \otimes \mathsf{I}_5 \right) \boldsymbol{w}_{\mathrm{R}_f} \right. \\ &\left. - \left(\boldsymbol{e}_1^{\mathrm{R}_f} \left(\boldsymbol{e}_N^{\mathrm{L}} \right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \mathsf{I}^2_{\mathrm{LtoR_f}} \otimes \mathsf{P}_{\mathrm{R}_f}^{(1D)} \mathsf{I}^3_{\mathrm{LtoR_f}} \otimes \mathsf{I}_5 \right) \boldsymbol{w}_{\mathrm{L}} \right\} + \boldsymbol{SAT}_{\mathrm{R}_f}^2,\end{aligned} \tag{66}$$

and $\boldsymbol{SAT}_{\mathrm{R}_f}^1$ and $\boldsymbol{SAT}_{\mathrm{R}_f}^2$ contain the SATs for the remaining five faces on element R_f.

7 Interface dissipation and boundary SATs

In order to render the entropy conservative scheme entropy stable, interface dissipation is added. The numerical dissipation added for the inviscid SATs (i.e., added to the right-hand side of the discretization) is motivated by a Roe approximate Riemann solver (for a detailed discussion see Refs. [5,43,20,18]). The inviscid dissipation for element L is given as

$$\boldsymbol{diss}_{\mathrm{L}} \equiv -\left(\mathsf{P}^{\mathrm{L}}\right)^{-1} \mathsf{R}_{\mathrm{L}}^{\mathrm{T}} \mathsf{P}^{\mathrm{L}}_{\perp\hat{\xi}_1} \left\{ \sum_{f=1}^{4} \mathsf{I}_{\mathrm{R}_f \mathrm{toL}} \left| \frac{\partial \boldsymbol{\mathcal{F}}^{I}_{\hat{\xi}_1}}{\partial \boldsymbol{\mathcal{W}}} \right|_{\mathrm{R}_f} \left(\mathsf{I}_{\mathrm{LtoR_f}} \mathsf{R}_{\mathrm{L}} \boldsymbol{w}_{\mathrm{L}} - \mathsf{R}_{\mathrm{R}_f} \boldsymbol{w}_{\mathrm{R}_f} \right) \right\}, \tag{67}$$

where

$$\begin{aligned}
\mathsf{R}_{\mathrm{L}} &\equiv \left(\boldsymbol{e}^{\mathrm{L}}_{N_1}\right)^{\mathrm{T}} \otimes \mathsf{I}_{\mathrm{L}} \otimes \mathsf{I}_{\mathrm{L}} \otimes \mathsf{I}_5, \\
\mathsf{P}^{\mathrm{L}}_{\perp\hat{\xi}_1} &\equiv \frac{\Delta^{\mathrm{L}}_2 \Delta^{\mathrm{L}}_3}{4} \mathsf{P}^{(1D)}_{\mathrm{L}} \otimes \mathsf{P}^{(1D)}_{\mathrm{L}} \otimes \mathsf{I}_5, \\
\mathsf{I}_{\mathrm{R}_f \mathrm{toL}} &\equiv \mathsf{I}^2_{\mathrm{R}_f \mathrm{toL}} \otimes \mathsf{I}^3_{\mathrm{R}_f \mathrm{toL}} \otimes \mathsf{I}_5, \\
\mathsf{I}_{\mathrm{LtoR_f}} &\equiv \mathsf{I}^2_{\mathrm{LtoR_f}} \otimes \mathsf{I}^3_{\mathrm{LtoR_f}} \otimes \mathsf{I}_5, \\
\mathsf{R}_{\mathrm{R}_f} &\equiv \left(\boldsymbol{e}^{\mathrm{R}_f}_{1}\right)^{\mathrm{T}} \otimes \mathsf{I}_{\mathrm{R}_f} \otimes \mathsf{I}_{\mathrm{R}_f} \otimes \mathsf{I}_5.
\end{aligned}$$

The inviscid dissipation term for the R_f element is constructed as

$$\boldsymbol{diss}_{\mathrm{R}_f} \equiv -\left(\mathsf{P}^{\mathrm{R}}_{f}\right)^{-1} \mathsf{R}^{\mathrm{T}}_{\mathrm{R}_f} \mathsf{P}^{\mathrm{R}_f}_{\perp\hat{\xi}_1} \left| \frac{\partial \boldsymbol{\mathcal{F}}^{I}_{\hat{\xi}_1}}{\partial \boldsymbol{\mathcal{W}}} \right|_{\mathrm{R}_f} \left(\mathsf{R}_{\mathrm{R}_f} \boldsymbol{w}_{\mathrm{R}_f} - \mathsf{I}_{\mathrm{LtoR_f}} \mathsf{R}_{\mathrm{L}} \boldsymbol{w}_{\mathrm{L}} \right), \tag{68}$$

where

$$\left| \frac{\partial \boldsymbol{\mathcal{F}}^{I}}{\partial \boldsymbol{\mathcal{W}}} \right| \equiv \mathsf{Y} \left| \Lambda \right| \mathsf{Y}^{\mathrm{T}}.$$

The matrices Y and Λ are block diagonal matrices constructed by assembling the pointwise 5×5 matrices obtained from the decomposition of the Jacobian matrix of $\boldsymbol{\mathcal{F}}$ with respect to $\boldsymbol{\mathcal{W}}$ and evaluated at the Roe-averaged of two states. In particular, $\left| \frac{\partial \boldsymbol{\mathcal{F}}^{I}_{\hat{\xi}_1}}{\partial \boldsymbol{\mathcal{W}}} \right|_{\mathrm{R}_f}$ is constructed from the Roe averaged states of $\mathsf{I}_{\mathrm{LtoR}_{\mathrm{q_L}}}$ and $\boldsymbol{q}_{\mathrm{R}_f}$.

Next, a theorem on the accuracy, stability, elementwise conservation, and free-stream preservation of the inviscid dissipation term is presented.

Theorem 5 *The dissipation terms Eqs. (67) and (68) are of order* $\min\left(p_{\mathrm{L}}, p_{\mathrm{R}_1}, \ldots, p_{\mathrm{R}_4}\right) + 3$*, and lead to an entropy stable inviscid scheme and have no impact on element-wise conservation or free-stream preservation.*

Proof The proofs are similar to those in Refs. [20,18] and are omitted for brevity.

The viscous interface dissipation terms (interior penalty terms) take the following form:

$$\boldsymbol{I}^{\mathrm{L}}_{\mathrm{P}} \equiv -\left(\mathsf{P}^{\mathrm{L}}\right)^{-1} \mathsf{R}^{\mathrm{T}}_{\mathrm{L}} \mathsf{P}^{\mathrm{L}}_{\perp\hat{\xi}_1} \mathsf{I}_{\mathrm{R}_f \mathrm{toL}} \tilde{\mathsf{J}}^{-1}_{\mathrm{R}_f} \tilde{\mathsf{C}}^{\mathrm{R}_f}_{1,1} \left(\mathsf{I}_{\mathrm{LtoR_f}} \mathsf{R}_{\mathrm{L}} \boldsymbol{w}_{\mathrm{L}} - \mathrm{R}_f \boldsymbol{w}_{\mathrm{R}_f} \right), \tag{69}$$

$$\boldsymbol{I}_{\mathrm{P}}^{\mathrm{R}_f} \equiv -\left(\mathsf{P}^{\mathrm{R}_f}\right)^{-1} \mathsf{R}_{\mathrm{R}_f}^{\mathrm{T}} \mathsf{P}_{\perp\hat{\xi}_1}^{\mathrm{R}_f} \tilde{\mathsf{J}}_{\mathrm{R}_f}^{-1} \tilde{\mathsf{C}}_{1,1}^{\mathrm{R}_f} \left(\mathsf{R}_{\mathrm{R}_f} \boldsymbol{w}_{\mathrm{R}_f} - \mathsf{I}_{\mathrm{LtoR_f}} \mathsf{R}_{\mathrm{L}} \boldsymbol{w}_{\mathrm{L}}\right), \tag{70}$$

where

$$\tilde{\mathsf{C}}_{1,1}^{\mathrm{R}_f} \equiv \frac{1}{2}\left\{\hat{\mathsf{C}}_{1,1}\left(\mathsf{I}_{\mathrm{LtoR_f}} \mathsf{R}_{\mathrm{L}} \boldsymbol{q}_{\mathrm{L}}\right) + \hat{\mathsf{C}}_{1,1}\left(\mathsf{R}_{\mathrm{R}_f} \boldsymbol{q}_{\mathrm{R}_f}\right)\right\},$$

and the diagonal matrix $\tilde{\mathsf{J}}_{\mathrm{R}_f}$ has the metric Jacobian associated with surface of element R_f along its diagonal. The next theorem summarizes the properties of the viscous dissipation terms.

Theorem 6 *The dissipation terms Eqs.* (69) *and* (70) *are of order* $\min\left(p_{\mathrm{L}}, p_{\mathrm{R}_1}, \ldots, p_{\mathrm{R}_f}\right)+3$*, lead to an entropy stable viscous scheme and have no impact on free-stream preservation.*

Proof The proofs are similar to those in Ref. [19] and are omitted for brevity.

In Section 8, four problems are used to characterize the h/p nonconforming algorithm: 1) the propagation of an isentropic vortex, 2) the propagation of a viscous shock, 3) the Taylor–Green vortex problem, and 4) the turbulent flow past a sphere. In all cases, the boundary conditions are weakly imposed by using the same type of mechanics as for the interface SATs discussed in this section (for details see Refs. [44,16]).

8 Numerical experiments

In this section, we verify that the proposed h/p-algorithm retains the accuracy and robustness of the conforming algorithm [5,44,11].

The unstructured grid h/p-adaptive solver used herein is developed at KAUST in the Advanced Algorithms and Numerical Simulations Laboratory, which is part of Extreme Computing Research Center (ECRC). It is built on top of the Portable and Extensible Toolkit for Scientific computing (PETSc) [3], its mesh topology abstraction (DMPLEX) [36] and scalable ordinary differential equation (ODE)/differential algebraic equations (DAE) solver library [1]. The p-refinement algorithm is fully implemented in the unstructured solver, whereas the h-refinement strategy leverages the capabilities of the p4est library [4]. Additionally, the conforming numerical solver is based on the algorithms proposed in Refs. [5,44,11]. The systems of ODEs arising from the spatial discretizations are integrated using the fourth-order accurate Dormand–Prince method [23] endowed with an adaptive time stepping technique based on digital signal processing [50,51]. To make the temporal error negligible, a tolerance of 10^{-8} is always used for the time-step adaptivity.

The errors are computed using volume scaled (for the L^1 and L^2 norms) discrete norms as follows:

$$\|\boldsymbol{u}\|_{L^1} = \Omega_c^{-1} \sum_{\kappa=1}^{K} \mathbf{1}_{\kappa}^{\mathrm{T}} \mathsf{P}^{\kappa} \mathsf{J}_{\kappa} \mathrm{abs}\left(\boldsymbol{u}_{\kappa}\right),$$

$$\|\boldsymbol{u}\|_{L^2}^2 = \Omega_c^{-1} \sum_{\kappa=1}^{K} \boldsymbol{u}_{\kappa} \mathsf{P}^{\kappa} \mathsf{J}_{\kappa} \boldsymbol{u}_{\kappa},$$

$$\|\boldsymbol{u}\|_{L^\infty} = \max_{\kappa=1\ldots K} \mathrm{abs}\left(\boldsymbol{u}_{\kappa}\right),$$

where Ω_c is the volume of Ω computed as $\Omega_c \equiv \sum_{\kappa=1}^{K} \mathbf{1}_\kappa^{\mathrm{T}} \mathsf{P}^\kappa \mathsf{J}_\kappa \mathbf{1}_\kappa$.

8.1 Isentropic Euler vortex propagation

For verification and characterization of the inviscid components of the algorithm, the propagation of an isentropic vortex is used. This benchmark problem has an analytical solution, which is given by

$$\mathcal{G}(x_1, x_2, x_3, t) = 1 - \left\{ [(x_1 - x_{1,0}) - U_\infty \cos(\alpha)\, t]^2 + [(x_2 - x_{2,0}) - U_\infty \sin(\alpha)\, t]^2 \right\},$$

$$\rho = T^{\frac{1}{\gamma - 1}},$$

$$\mathcal{U}_1 = U_\infty \cos(\alpha) - \epsilon_\nu \frac{(x_2 - x_{2,0}) - U_\infty \sin(\alpha)\, t}{2\pi} \exp\left(\frac{\mathcal{G}}{2}\right),$$

$$\mathcal{U}_2 = U_\infty \sin(\alpha) - \epsilon_\nu \frac{(x_1 - x_{1,0}) - U_\infty \cos(\alpha)\, t}{2\pi} \exp\left(\frac{\mathcal{G}}{2}\right),$$

$$\mathcal{U}_3 = 0,$$

$$T = \left[1 - \epsilon_\nu^2 M_\infty^2 \frac{\gamma - 1}{8\pi^2} \exp(\mathcal{G})\right],$$

where U_∞, M_∞, and $(x_{1,0}, x_{2,0}, x_{3,0})$ are the modulus of the free-stream velocity, the free-stream Mach number, and the vortex center, respectively. In this paper, the following values are used: $U_\infty = M_\infty c_\infty$, $\epsilon_\nu = 5$, $M_\infty = 0.5$, $\gamma = 1.4$, $\alpha = 45^\circ$, and $(x_{1,0}, x_{2,0}, x_{3,0}) = (0, 0, 0)$. The computational domain is

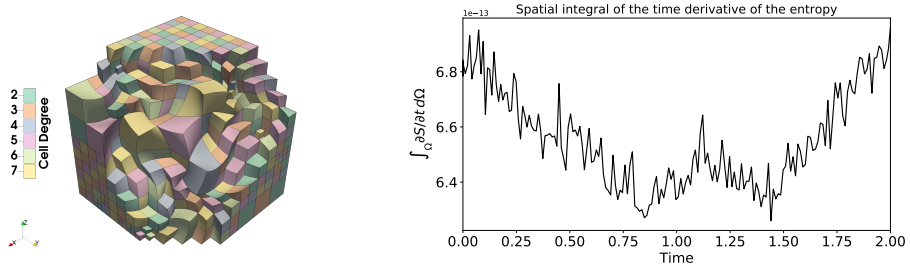
$$x_1 \in [-5, 5], \qquad x_2 \in [-5, 5], \qquad x_3 \in [-5, 5], \qquad t \in [0, 2].$$

The analytical solution is used to furnish data for the initial condition.

First, we report on the results aimed at validating the entropy conservation properties of the interior domain SBP-SAT algorithm. Thus, periodic boundary conditions are used on all six faces of the computational domain. Furthermore, all the dissipation terms used for the interface coupling are turned off. The discrete integral over the volume of the time rate of change of the entropy function, $\int_\Omega \frac{\partial \mathcal{S}}{\partial t} \mathrm{d}\Omega$, is monitored at every time step. This means that at each time step the compressible Euler equations are multiplied by the discrete entropy variables to construct the discrete analog of the right-hand side of Eq. (55).

We subdivide the computational domain using ten hexahedrons in each coordinate direction. Subsequently, we split random cells in the mesh using one or two levels of h-refinement. Then, we assign the solution polynomial degree in each element to a random integer chosen uniformly from the set $\{2, 3, 4, 5\}$ (i.e., each member in the set has an equal probability of being chosen). To test the conservation of entropy and therefore the free-stream condition when curved element interfaces are used, we construct the LGL collocation point coordinates at element interfaces[1] as follows:

[1] In a general setting, element interfaces can also be boundary element interfaces.



(a) Polynomial degree distribution. (b) Time rate of change of the entropy function.

Fig. 3: Isentropic Euler vortex.

– Construct a mesh using a p_ith-order polynomial approximation for the element interfaces.
– Perturb the nodes that are used to define the p_ith-order polynomial approximation of the element interfaces as follows:

$$x_1 = x_{1,*} + \frac{1}{15} L_1 \cos(a) \cos(3b) \sin(4c),$$
$$x_2 = x_{2,*} + \frac{1}{15} L_2 \sin(4a) \cos(b) \cos(3c),$$
$$x_3 = x_{3,*} + \frac{1}{15} L_3 \cos(3a) \sin(4b) \cos(c),$$

where,

$$a = \frac{\pi}{L_1} \left(x_{1,*} - \frac{x_{1,H} + x_{1,L}}{2} \right), \quad b = \frac{\pi}{L_2} \left(x_{2,*} - \frac{x_{2,H} + x_{2,L}}{2} \right),$$
$$c = \frac{\pi}{L_3} \left(x_{3,*} - \frac{x_{3,H} + x_{3,L}}{2} \right).$$

The symbols L_1, L_2 and L_3 represent the dimensions of the computational domain in the three coordinate directions and the subscript $*$ indicates the unperturbed coordinate of the nodes. This step yields a perturbed p_ith-order polynomial.
– Compute the coordinate of the LGL points at the element interface by evaluating the perturbed p_ith-order polynomial at the LGL points used to define the cell solution polynomial of order p_s.

Herein, we use $p_i = 2$. Figure 3a shows a cut of the mesh where each cell is colored according to the solution polynomial degree assigned to it. Curved element interfaces are clearly visible.

The propagation of the vortex is simulated for two time units. Figure 3b plots the integral over the volume of the time derivative of the entropy function. We can see that the global variation of the discrete time rate of change of $\mathcal{S}$ is practically zero (i.e., machine double precision). This implies that the nonconforming algorithm is entropy conservative.

Second, we perform a grid convergence study to investigate the order of convergence of the h/p-adaptive approach. The base grid (labeled with "0" in the first column of following tables) is constructed as follows:

- Divide the computational domain into four hexahedral elements in each coordinate direction.
- Refine random elements by using one or two levels of h-refinement.
- Assign the solution polynomial degree in each element to a random integer chosen uniformly from the set $\{p_s, p_s + 1\}$.
- Approximate the curved element interfaces with a p_sth-order accurate polynomial.
- Construct the perturbed elements and their corresponding LGL points as described previously.

From the base grid, which is similar to the one depicted in Figure 3a, a sequence of nested grids is then generated to perform the convergence study. The results are reported in Tables 1 through 4 for the error on the density. The number listed in the first column denoted by "Levels" indicates the number of uniform refinements in each coordinate direction.

	Conforming, $p = 1$						Nonconforming, $p = 1$ and $p = 2$					
Levels	L^1	Rate	L^2	Rate	L^∞	Rate	L^1	Rate	L^2	Rate	L^∞	Rate
0	2.74E-02	-	1.32E-03	-	1.55E-01	-	1.02E-02	-	5.80E-04	-	1.43E-01	-
1	1.14E-02	-1.26	6.61E-04	-1.00	1.12E-01	-0.47	4.38E-03	-1.22	2.82E-04	-1.04	7.17E-02	-1.00
2	5.13E-03	-1.16	3.31E-04	-1.00	7.29E-02	-0.62	1.45E-03	-1.60	9.99E-05	-1.50	4.30E-02	-0.74
3	1.70E-03	-1.59	1.15E-04	-1.52	3.01E-02	-1.28	4.16E-04	-1.80	3.02E-05	-1.72	2.29E-02	-0.91
4	4.76E-04	-1.84	3.24E-05	-1.83	8.53E-03	-1.82	1.11E-04	-1.91	8.76E-06	-1.79	1.06E-02	-1.10
5	1.23E-04	-1.96	8.33E-06	-1.96	2.13E-03	-2.00	2.61E-05	-2.09	2.28E-06	-1.94	4.05E-03	-1.40
6	3.08E-05	-1.99	2.09E-06	-1.99	5.24E-04	-2.02	6.31E-06	-2.05	5.75E-07	-1.99	1.25E-03	-1.70
7	7.68E-06	-2.00	5.23E-07	-2.00	1.30E-04	-2.01	1.56E-06	-2.02	1.50E-07	-1.94	4.45E-04	-1.48

Table 1: Convergence study of the isentropic vortex propagation: two levels of h-refinement, $p = 1$ and $p = 2$; density error.

	Conforming, $p = 2$						Nonconforming, $p = 2$ and $p = 3$					
Levels	L^1	Rate	L^2	Rate	L^∞	Rate	L^1	Rate	L^2	Rate	L^∞	Rate
0	9.75E-03	-	5.32E-04	-	1.24E-01	-	2.59E-03	-	1.75E-04	-	7.52E-02	-
1	3.18E-03	-1.61	2.09E-04	-1.35	6.97E-02	-0.83	4.11E-04	-2.65	3.45E-05	-2.34	3.14E-02	-1.26
2	5.18E-04	-2.62	3.88E-05	-2.43	2.51E-02	-1.47	4.91E-05	-3.07	5.13E-06	-2.75	8.16E-03	-1.95
3	6.38E-05	-3.02	5.50E-06	-2.82	7.23E-03	-1.79	5.86E-06	-3.07	6.74E-07	-2.93	2.09E-03	-1.97
4	7.61E-06	-3.07	7.23E-07	-2.93	1.21E-03	-2.58	7.05E-07	-3.06	8.97E-08	-2.91	4.32E-04	-2.27
5	9.48E-07	-3.00	9.95E-08	-2.86	2.75E-04	-2.14	8.54E-08	-3.04	1.20E-08	-2.91	1.00E-04	-2.11
6	1.23E-07	-2.94	1.43E-08	-2.83	3.41E-05	-3.01	9.98E-09	-3.10	1.56E-09	-2.94	2.52E-05	-1.99

Table 2: Convergence study of the isentropic vortex propagation: two levels of h-refinement, $p = 2$ and $p = 3$; density error.

For all the degrees tested (i.e., $p = 1$ to $p = 5$), the order of convergence of the conforming and nonconforming algorithms is very similar. However, note that

	Conforming, $p = 3$						Nonconforming, $p = 3$ and $p = 4$					
Levels	L^1	Rate	L^2	Rate	L^∞	Rate	L^1	Rate	L^2	Rate	L^∞	Rate
0	5.22E-03	-	3.38E-04	-	9.16E-02	-	5.38E-04	-	4.51E-05	-	5.38E-02	-
1	6.84E-04	-2.93	5.30E-05	-2.67	4.71E-02	-0.96	4.18E-05	-3.69	3.89E-06	-3.54	5.42E-03	-3.31
2	5.50E-05	-3.64	4.54E-06	-3.55	6.61E-03	-2.83	2.61E-06	-4.00	2.82E-07	-3.78	6.47E-04	-3.07
3	3.48E-06	-3.98	3.33E-07	-3.77	5.47E-04	-3.59	1.79E-07	-3.86	1.92E-08	-3.88	7.36E-05	-3.14
4	2.10E-07	-4.05	2.45E-08	-3.76	4.93E-05	-3.47	1.09E-08	-4.04	1.23E-09	-3.96	6.93E-06	-3.41
5	1.39E-08	-3.92	1.87E-09	-3.71	6.32E-06	-2.96	7.05E-10	-3.96	8.10E-11	-3.93	8.10E-07	-3.10

Table 3: Convergence study of the isentropic vortex propagation: two levels of h-refinement, $p = 3$ and $p = 4$; density error.

	Conforming, $p = 4$						Nonconforming, $p = 4$ and $p = 5$					
Levels	L^1	Rate	L^2	Rate	L^∞	Rate	L^1	Rate	L^2	Rate	L^∞	Rate
0	2.34E-03	-	1.56E-04	-	9.92E-02	-	7.48E-05	-	5.15E-06	-	4.21E-03	-
1	1.70E-04	-3.78	1.43E-05	-3.45	2.32E-02	-2.09	2.20E-06	-5.09	1.87E-07	-4.79	2.45E-04	-4.10
2	6.07E-06	-4.81	6.41E-07	-4.48	1.27E-03	-4.19	6.76E-08	-5.02	6.66E-09	-4.81	1.81E-05	-3.76
3	1.99E-07	-4.93	2.25E-08	-4.83	5.13E-05	-4.64	2.08E-09	-5.03	2.21E-10	-4.91	1.26E-06	-3.84
4	7.11E-09	-4.81	8.60E-10	-4.71	3.01E-06	-4.09	6.61E-11	-4.97	6.78E-12	-5.03	9.13E-08	-3.79

Table 4: Convergence study of the isentropic vortex propagation: two levels of h-refinement, $p = 4$ and $p = 5$; density error.

in the L^1 and L^2 norms, the nonconforming algorithm is more accurate than the conforming one. In the discrete L^∞ norm, the nonconforming scheme is sometimes slightly worse than the conforming scheme; this most likely results from the fact that the interpolation matrices are suboptimal.

8.2 Viscous shock propagation

Next we study the propagation of a viscous shock using the compressible Navier–Stokes equations. We assume a planar shock propagating along the x_1 coordinate direction with a Prandtl number of $Pr_\infty = 3/4$. Note that the subscript ∞ refers to the undisturbed flow on the the high pressure side of the viscous shock. The exact solution of this problem is known; the momentum $\mathcal{V}(x_1)$ satisfies the ODE

$$\alpha \mathcal{V} \frac{\partial \mathcal{V}}{\partial x_1} - (\mathcal{V} - 1)(\mathcal{V} - \mathcal{V}_f) = 0; \ -\infty \leq x_1 \leq +\infty, \ t \geq 0, \tag{71}$$

whose solution can be written implicitly as

$$\begin{aligned} x_1 - \frac{1}{2}\alpha \Big(& \log \left| (\mathcal{V}(x_1) - 1)(\mathcal{V}(x_1) - \mathcal{V}_f) \right| \\ & + \frac{1 + \mathcal{V}_f}{1 - \mathcal{V}_f} \log \left| \frac{\mathcal{V}(x_1) - 1}{\mathcal{V}(x_1) - \mathcal{V}_f} \right| \Big) = 0, \end{aligned} \tag{72}$$

where

$$\mathcal{V}_f \equiv \frac{\mathcal{U}_L}{\mathcal{U}_R}, \qquad \alpha \equiv \frac{2\gamma}{\gamma + 1} \frac{\mu}{Pr_\infty \dot{\mathcal{M}}}. \tag{73}$$

Here, $\mathcal{U}_{L/R}$ are known velocities to the left and right of the shock at $-\infty$ and $+\infty$, respectively, $\dot{\mathcal{M}}$ is the constant mass flow across the shock, Pr is the Prandtl number, and μ is the dynamic viscosity. The mass and total enthalpy are constant across the shock. Moreover, the momentum and energy equations become redundant.

For our tests, $\mathcal{V}$ is computed from Equation (72) to machine precision using bisection. The moving shock solution is obtained by applying a uniform translation to the above solution. The shock is located at the center of the domain at $t = 0$ and the following values are used: $M_\infty = 2.5$, $Re_\infty = 10$, and $\gamma = 1.4$. The domain is given by

$$x_1 \in [-0.5, 0.5], \quad x_2 \in [-0.5, 0.5], \quad x_3 \in [-0.5, 0.5], \quad t \in [0, 0.5].$$

The boundary conditions are prescribed by penalizing the numerical solution against the exact solution. The analytical solution is also used to furnish data for the initial condition.

The base grid (labeled with "0" in the first column of Tables 5 through 8) is constructed as described in Section 8.1. From the base grid, which is similar to the one depicted in Figure 3a, a sequence of nested grids is then generated to perform the convergence study. The results are reported in Tables 5 through 8 for the error on the density. Again, the number listed in the first column denoted by "Levels" indicates the number of uniform refinement in each coordinate direction.

Similar to the propagation of the inviscid vortex, for all the degrees tested (i.e., $p = 1$ to $p = 5$), the order of convergence of the conforming and nonconforming algorithms is similar. However, note that, the nonconforming algorithm is more accurate than the conforming algorithm, for all the three norms reported.

	Conforming, $p = 1$						Nonconforming, $p = 1$ and $p = 2$					
Levels	L^1	Rate	L^2	Rate	L^∞	Rate	L^1	Rate	L^2	Rate	L^∞	Rate
0	5.43E-02	-	6.54E-02	-	1.40E-01	-	1.31E-02	-	2.11E-02	-	7.32E-02	-
1	2.04E-02	-1.41	2.92E-02	-1.16	8.42E-02	-0.73	3.29E-03	-1.99	5.76E-03	-1.87	2.94E-02	-1.32
2	5.56E-03	-1.87	8.45E-03	-1.79	2.85E-02	-1.57	8.39E-04	-1.97	1.46E-03	-1.98	9.16E-03	-1.68
3	1.44E-03	-1.94	2.23E-03	-1.92	8.12E-03	-1.81	2.11E-04	-1.99	3.76E-04	-1.96	2.36E-03	-1.96
4	3.68E-04	-1.97	5.66E-04	-1.98	2.26E-03	-1.84	5.03E-05	-2.07	8.97E-05	-2.07	5.97E-04	-1.98
5	9.28E-05	-1.99	1.43E-04	-1.99	6.05E-04	-1.90	1.24E-05	-2.02	2.10E-05	-2.09	1.48E-04	-2.01

Table 5: Convergence study of the viscous shock propagation: two levels of h-refinement, $p = 1$ and $p = 2$; density error.

8.3 Taylor–Green vortex at $Re = 1,600$

The purpose of this section is to demonstrate that the nonconforming algorithm has the same stability properties as the conforming algorithm. To do so, the Taylor–Green vortex problem on a very coarse grid is solved.

	Conforming, $p = 2$						Nonconforming, $p = 2$ and $p = 3$					
Levels	L^1	Rate	L^2	Rate	L^∞	Rate	L^1	Rate	L^2	Rate	L^∞	Rate
0	1.78E-02	-	2.68E-02	-	1.36E-01	-	1.85E-03	-	3.84E-03	-	4.86E-02	-
1	2.93E-03	-2.60	5.05E-03	-2.41	5.98E-02	-1.19	2.75E-04	-2.75	5.85E-04	-2.72	1.11E-02	-2.14
2	3.86E-04	-2.92	6.93E-04	-2.87	1.09E-02	-2.45	4.01E-05	-2.78	8.74E-05	-2.74	2.03E-03	-2.45
3	5.55E-05	-2.80	1.03E-04	-2.74	2.23E-03	-2.29	5.00E-06	-3.00	1.01E-05	-3.11	3.18E-04	-2.67
4	8.96E-06	-2.63	1.79E-05	-2.53	4.96E-04	-2.17	6.10E-07	-3.04	1.23E-06	-3.04	4.20E-05	-2.92
5	1.46E-06	-2.66	2.99E-06	-2.58	8.96E-05	-2.47	7.00E-08	-3.12	1.51E-07	-3.03	5.50E-06	-2.93

Table 6: Convergence study of the viscous shock propagation: two levels of h-refinement, $p = 2$ and $p = 3$; density error.

	Conforming, $p = 3$						Nonconforming, $p = 3$ and $p = 4$					
Levels	L^1	Rate	L^2	Rate	L^∞	Rate	L^1	Rate	L^2	Rate	L^∞	Rate
0	4.45E-03	-	7.52E-03	-	7.51E-02	-	2.41E-04	-	5.24E-04	-	1.12E-02	-
1	3.40E-04	-3.71	6.50E-04	-3.53	1.19E-02	-2.66	1.80E-05	-3.74	4.11E-05	-3.67	1.01E-03	-3.47
2	2.67E-05	-3.67	5.36E-05	-3.60	1.20E-03	-3.30	1.21E-06	-3.90	3.00E-06	-3.78	8.17E-05	-3.63
3	1.95E-06	-3.77	4.25E-06	-3.66	1.25E-04	-3.26	7.30E-08	-4.05	1.90E-07	-3.98	5.82E-06	-3.81
4	1.48E-07	-3.72	3.67E-07	-3.53	1.12E-05	-3.48	4.23E-09	-4.11	1.09E-08	-4.13	3.60E-07	-4.02

Table 7: Convergence study of the viscous shock propagation: two levels of h-refinement, $p = 3$ and $p = 4$; density error.

	Conforming, $p = 4$						Nonconforming, $p = 4$ and $p = 5$					
Levels	L^1	Rate	L^2	Rate	L^∞	Rate	L^1	Rate	L^2	Rate	L^∞	Rate
0	1.21E-03	-	2.28E-03	-	2.50E-02	-	3.47E-05	-	8.15E-05	-	1.90E-03	-
1	8.50E-05	-3.83	1.54E-04	-3.88	3.04E-03	-3.04	1.37E-06	-4.66	3.13E-06	-4.70	8.02E-05	-4.57
2	2.75E-05	-4.95	5.66E-06	-4.77	1.52E-04	-4.32	4.73E-08	-4.85	1.10E-07	-4.83	3.04E-06	-4.72
3	1.16E-07	-4.57	2.54E-07	-4.48	7.54E-06	-4.33	1.62E-09	-4.87	3.62E-09	-4.93	1.44E-07	-4.40
4	5.21E-09	-4.48	1.11E-08	-4.52	4.01E-07	-4.23	6.01E-11	-4.75	1.30E-10	-4.80	6.96E-09	-4.37

Table 8: Convergence study of the viscous shock propagation: two levels of h-refinement, $p = 4$ and $p = 5$; density error.

The numerical solution is computed in a periodic cube $[-\pi L \leq x, y, z \leq \pi L]$ and the initial condition is given by

$$\begin{aligned}
&\mathcal{U}_1 = \mathcal{V}_0 \sin\left(\frac{x_1}{L}\right) \cos\left(\frac{x_2}{L}\right) \cos\left(\frac{x_3}{L}\right), \\
&\mathcal{U}_2 = -\mathcal{V}_0 \cos\left(\frac{x_1}{L}\right) \sin\left(\frac{x_2}{L}\right) \cos\left(\frac{x_3}{L}\right), \\
&\mathcal{U}_3 = 0, \\
&\mathcal{P} = \mathcal{P}_0 + \frac{\rho_0 \mathcal{V}_0^2}{16} \left[\cos\left(\frac{2x_1}{L} + \cos\left(\frac{2x_2}{L}\right)\right)\right] \left[\cos\left(\frac{2x_3}{L} + 2\right)\right].
\end{aligned} \tag{74}$$

The flow is initialized to be isothermal, i.e., $\mathcal{P}/\rho = \mathcal{P}_0/\rho_0 = R\mathcal{T}_0$, and $\mathcal{P}_0 = 1$, $\mathcal{T}_0 = 1$, $L = 1$, and $\mathcal{V}_0 = 1$. Finally, the Reynolds number is defined by $Re = (\rho_0 \mathcal{V}_0)/\mu$, where μ is the dynamic viscosity.

To obtain results that are reasonably close to those found for the incompressible equations, a Mach number of $M = 0.05$ is used. Furthermore, the Reynolds number, the Prandtl number, and the initial density distribution are set to $Re = 1600$, $Pr = 0.71$, and $\rho_0 = \gamma M^2$, respectively, where $\gamma = 1.4$.

For this test case, a grid is constructed as follows:

- Divide the computational domain with N_{1h} hexahedral elements in each coordinate direction.
- Refine random elements by using randomly one or two levels of h-refinement.
- Assign the solution polynomial degree in each element to a random integer chosen uniformly from a set (see the legend in Figure 4).
- Construct the perturbed elements and their corresponding LGL points as described previously (the element interfaces are approximated using a polynomial degree which is the minimum solution polynomial degree used in the simulation).

Herein, two grids with $N_{1h} = 4$ and $N_{1h} = 8$ are considered. The total number of hexahedrons is 869 and 7547, respectively. The simulations are run without additional stabilization mechanisms (dissipation model, dealiasing, filtering, etc.), where the only numerical dissipation originates from the upwind interelement coupling procedure.

Figure 4 shows the time rate of change of the kinetic energy, dke/dt, for the nonconforming algorithm using a random distribution of the solution polynomial order between i) $p = 2$ and $p = 8$ and ii) $p = 7$ and $p = 13$. The reference direct numerical solution (DNS) reported in [60] is also plotted. We note that by increasing the order of accuracy of the solution polynomial in each cell and the grid density, the solution moves closer to the DNS solution. The main takeaway from the figure is that all simulations are stable, which is numerical evidence that the h/p nonconforming scheme inherits the stability characteristics of the conforming and fully-staggered algorithms [5,44,6,11,42].

8.4 Flow around a sphere at $Re = 2000$

In this section, we test our implementation within a more complex setting represented by the flow around a sphere at $Re_\infty = 2000$, based on the sphere diameter, d, and $M_\infty = 0.05$. These similarity flow parameters are defined using the undisturbed flow conditions upstream of the sphere. With this value of the Reynolds number, the flow is fully turbulent.

The sphere is centered at the origin of the axes, and a box is respectively extended $20d$ and $60d$ upstream and downstream the direction of the flow; the box size is $30d$ in both the x_2 and x_3 directions. As boundary conditions, we consider adiabatic solid walls at the surface of the sphere [16] and far field on all faces of the box. We use a grid with 24,704 hexahedral elements. Figure 5 shows the mesh near the sphere. The colors indicate the solution polynomial order used in each cell. The quality of the elements is good in the boundary layer region whereas in the other portion of the domain the quality is fairly poor. This choice is intentional and is

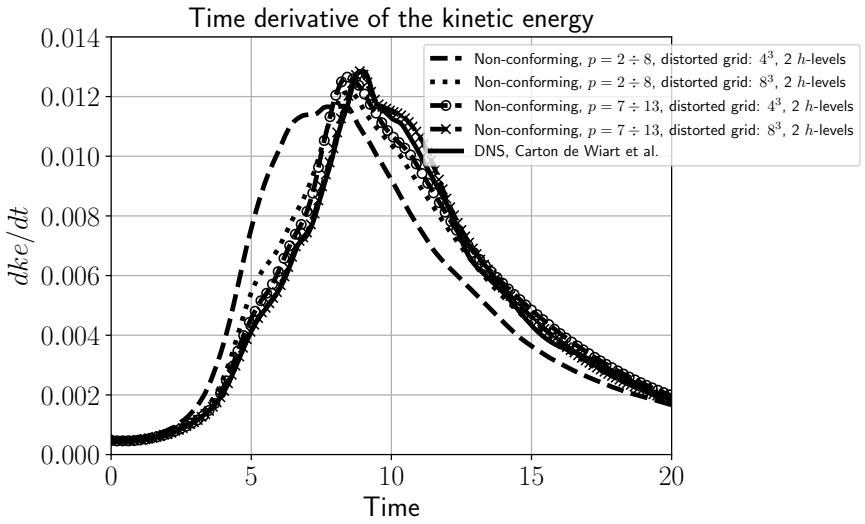


Fig. 4: Evolution of the time derivative of the kinetic energy for the Taylor–Green vortex at $Re = 1600$, $M = 0.05$.

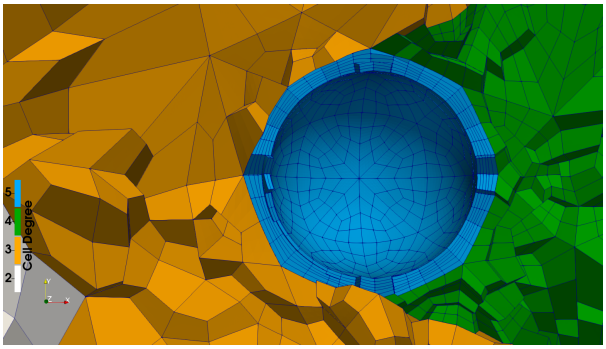


Fig. 5: Polynomial degree distribution for the mesh around a sphere; gray $p = 2$, orange $p = 3$, blue $p = 6$, and green $p = 4$.

	$\langle C_D \rangle$
Munson et al. [37]	0.412
Present	0.416

Table 9: Time-average drag coefficient of a sphere at $Re = 2000$, $M = 0.05$.

for the purpose of demonstrating the performance of the algorithm on nonideal grids.

We compute the time-average value of the drag coefficient, $\langle C_D \rangle$, and we compare it with the value reported in literature [37]. From Table 9, we can see that the computed time-average drag coefficient matches very well the reference value.

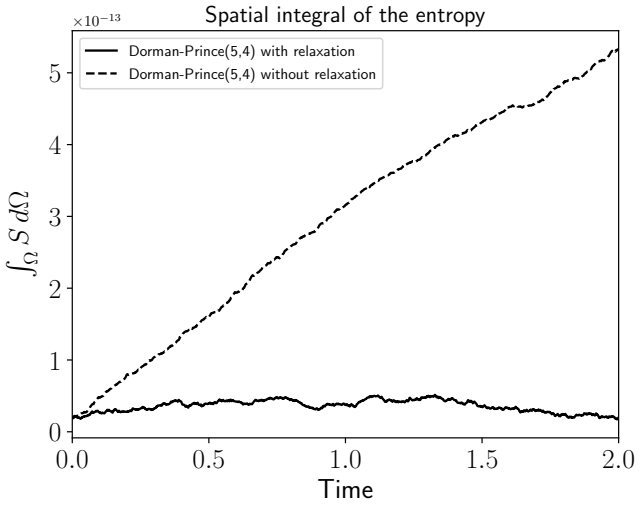


Fig. 6: Evolution of the discrete spatial integral of the entropy function.

8.5 Entropy conservation of the fully-discrete explicit discretization

To conclude the numerical results section, we demonstrate the entropy conservation of the fully-discrete explicit discretization of the compressible Navier–Stokes equations by integrating in time the system of ODEs, which arise from the spatial discretization with an explicit relaxation Runge–Kutta scheme [46]. As shown in Refs. [34,46], the term "relaxation" represents a general approach that allows any Runge–Kutta method to preserve the correct time evolution of an arbitrary functional, without sacrificing the linear covariance, accuracy, or stability properties of the original method. In the context of the compressible Euler and Navier–Stokes equations, the relaxation Runge–Kutta scheme is constructed to preserve the discrete entropy function obtained from the spatial discretization. This leads to a fully discrete algorithm that is entropy conservative or entropy stable if the spatial discretization is entropy conservative or entropy stable, respectively.

As a model problem, we again use the propagation of an isentropic vortex and we analyze the time evolution of the entropy function, which for the current setting must be zero. The same grid and solution polynomial distribution shown in Figure 3a is used for this test case. To achieve entropy conservation at the spatial level, all the dissipation terms used for the interface coupling are turned off, including upwind and interior penalty SATs. The fourth-order accurate Dormand–Prince method [23], with and without a relaxation algorithm, are used. We show the entropy variation in Figure 6. The entropy is conserved up to machine (double) precision using relaxation, whereas, without relaxation, the solution shows significant essentially monotone changes in the total entropy function.

9 Conclusions

In this paper, the p-refinement/coarsening algorithms in [19,20,18] are extended to arbitrary h/p-refinement/coarsening. To obtain an algorithm for which the discrete GCL conditions are solved for element by element, the surface metric terms need to be localized to the small elements on an h-refined face. The discrete GCL conditions are then solved using the procedure in Crean et al. [14]. The resulting algorithm is entropy conservative/stable, elementwise conservative, and free-stream preserving. Finally, the algorithm is shown to retain the accuracy and stability characteristics of the original conforming scheme [5,42] on a set of test problems and, when coupled with relaxation Runge–Kutta schemes [46], yields a fully discrete entropy conservative/stable scheme.

Acknowledgements The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST). We are thankful for the computing resources of the Supercomputing Laboratory and the Extreme Computing Research Center at KAUST. Special thanks are extended to Dr. Mujeeb R. Malik for supporting this work as part of the NASA Transformational Tools and Technologies, T^3, project.

References

1. Abhyankar, S., Brown, J., Constantinescu, E.M., Ghosh, D., Smith, B.F., Zhang, H.: PETSc/TS: A modern scalable ODE/DAE solver library. arXiv preprint arXiv:1806.01437 (2018)
2. Altmann, C., Beck, A.D., Hindenlang, F., Staudenmaier, M., Gassner, G.J., Munz, C.D.: An Efficient High Performance Parallelization of a Discontinuous Galerkin Spectral Element Method, pp. 37–47. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
3. Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W.D., Karpeyev, D., Kaushik, D., Knepley, M.G., May, D.A., McInnes, L.C., Mills, R.T., Munson, T., Rupp, K., Sanan, P., Smith, B.F., Zampini, S., Zhang, H., Zhang, H.: PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.11, Argonne National Laboratory (2019)
4. Burstedde, C., Wilcox, L.C., Ghattas, O.: `p4est`: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. SIAM Journal on Scientific Computing **33**(3), 1103–1133 (2011). DOI 10.1137/100791634
5. Carpenter, M.H., Fisher, T.C., Nielsen, E.J., Frankel, S.H.: Entropy stable spectral collocation schemes for the Navier–Stokes equations: discontinuous interfaces. SIAM Journal on Scientific Computing **36**(5), B835–B867 (2014)
6. Carpenter, M.H., Fisher, T.C., Nielsen, E.J., Parsani, M., Svärd, M., Yamaleev, N.: Entropy stable summation-by-parts formulations for computational fluid dynamics. Handbook of Numerical Analysis (17), 495–524 (2016)
7. Carpenter, M.H., Gottlieb, D., Abarbanel, S.: Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: Methodology and application to high-order compact schemes. Journal of Computational Physics **111**(2), 220–236 (1994)
8. Carpenter, M.H., Nordström, J., Gottlieb, D.: A stable and conservative interface treatment of arbitrary spatial accuracy. Journal of Computational Physics **148**(2), 341–365 (1999)
9. Carpenter, M.H., Nordström, J., Gottlieb, D.: Revisiting and extending interface penalties for multi-domain summation-by-parts operators. Journal of Scientific Computing **45**(1-3), 118–150 (2010)
10. Carpenter, M.H., Parsani, M., Fisher, T.C., Nielsen, E.J.: Entropy stable staggered grid spectral collocation for the Burgers' and compressible Navier–Stokes equations. NASA TM-2015-218990 (2015)
11. Carpenter, M.H., Parsani, M., Fisher, T.C., Nielsen, E.J.: Towards and entropy stable spectral element framework for computational fluid dynamics. In: 54th AIAA Aerospace

Sciences Meeting, AIAA 2016-1058. American Institute of Aeronautics and Astronautics (AIAA) (2016)
12. Chan, J., Del Rey Fernández, D.C., Carpenter, M.H.: Efficient entropy stable Gauss collocation methods. (Submitted to SIAM Journal on Scientific Computing) (2018)
13. Chandrashekar, P.: Kinetic energy preserving and entropy stable finite volume schemes for compressible Euler and Navier–Stokes equations. Communications in Computational Physics **14**(5), 1252–1286 (2013)
14. Crean, J., Hicken, J.E., Del Rey Fernández, D.C., Zingg, D.W., Carpenter, M.H.: Entropy-stable summation-by-parts discretization of the Euler equations on general curved elements. Journal of Computational Physics **356**, 410 –438 (2018)
15. Dafermos, C.M.: Hyperbolic conservation laws in continuum physics. Springer-Verlag, Berlin (2010)
16. Dalcin, L., Rojas, D., Zampini, S., Del Rey Fernández, D.C., Carpenter, M.H., Parsani, M.: Conservative and entropy stable solid wall boundary conditions for the compressible Navier–Stokes equations: Adiabatic wall and heat entropy transfer. Journal of Computational Physics **397** (2019)
17. Del Rey Fernández, D.C., Boom, P.D., Zingg, D.W.: A generalized framework for nodal first derivative summation-by-parts operators. Journal of Computational Physics **266**(1), 214–239 (2014)
18. Del Rey Fernández, D.C., Carpenter, M.H., Dalcin, L., Fredrich, L., Winters, A.R., Gassner, G.J., Zampini, S., Parsani, M.: Entropy stable non-conforming discretizations with the summation-by-parts property for curvilinear coordinates. NASA TM-2019- (2019)
19. Del Rey Fernández, D.C., Carpenter, M.H., Dalcin, L., Fredrich, L., Winters, A.R., Gassner, G.J., Zampini, S., Parsani, M.: Entropy stable nonconforming discretization with the summation-by-parts property for the compressible Navier–Stokes equations. Submitted Computers & fluids (2019)
20. Del Rey Fernández, D.C., Carpenter, M.H., Dalcin, L., Fredrich, L., Winters, A.R., Gassner, G.J., Zampini, S., Parsani, M.: Entropy stable $p-$nonconforming discretizations with the summation-by-parts property for the compressible Euler equations. Submitted SIAM Journal of Scientific Computing (2019)
21. Del Rey Fernández, D.C., Crean, J., Carpenter, M.H., Hicken, J.E.: Staggered entropy-stable summation-by-parts discretization of the Euler equations on general curved elements. Journal of Computational Physics **392**, 161–186 (2019)
22. Del Rey Fernández, D.C., Hicken, J.E., Zingg, D.W.: Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. Computers & Fluids **95**(22), 171–196 (2014)
23. Dormand, J.R., Prince, P.J.: A family of embedded Runge–Kutta formulae. Journal of Computational and Applied Mathematics **6**(1), 19 – 26 (1980)
24. Fisher, T.C.: High-order l^2 stable multi-domain finite difference method for compressible flows. Ph.D. thesis, Purdue University (2012)
25. Fisher, T.C., Carpenter, M.H.: High-order entropy stable finite difference schemes for nonlinear conservation laws: Finite domains. Journal of Computational Physics **252**(1), 518–557 (2013)
26. Fisher, T.C., Carpenter, M.H., Nordström, J., Yamaleev, N.K.: Discretely conservative finite-difference formulations for nonlinear conservation laws in split form: Theory and boundary conditions. Journal of Computational Physics **234**(1), 353–375 (2013)
27. Flad, D., Gassner, G.J.: On the use of kinetic energy preserving DG-schemes for large eddy simulation. Journal of Computational Physics **350**, 782–795 (2017)
28. Friedrich, L., Shnücke, G., Winters, A.R., Del Rey Fernández, D.C., Gassner, G.J., Carpenter, M.H.: Entropy stable space-time discontinuous Galerkin schemes with summation-by-parts property for hyperbolic conservation laws. Journal of Scientific Computing **80**(1), 175–222 (2019)
29. Friedrich, L., Winters, A.R., Del Rey Fernández, D.C., Gassner, G.J., Parsani, M., Carpenter, M.H.: An entropy stable h/p non-conforming discontinuous Galerkin method with the summation-by-parts property. Journal of Scientific Computing pp. 1–37 (2018)
30. Gassner, G.J., Winters, A.R., Kopriva, D.A.: Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations. Journal of Computational Physics **327**(C), 39–66 (2016)
31. Gassner, G.J., Winters, A.R., Kopriva, D.A.: A well balanced and entropy conservative discontinuous Galerkin spectral element method for the shallow water equations. Applied Mathematics and Computation **272**(2), 291–308 (2016)

32. Hadri, B., Parsani, M., Hutchinson, M., Heinecke, A., Dalcin, L., Keyes, D.: Performance study of sustained petascale direct numerical simulation on Cray XC40 systems (Trinity, Shaheen2 and Cori). Concurrency and Computation: Practice and Experience (2020)
33. Hutchinson, M., Heinecke, A., Pabst, H., Henry, G., Parsani, M., Keyes, D.: Efficiency of high order spectral element methods on petascale architectures. In: International Conference on High Performance Computing, pp. 449–466 (2016)
34. Ketcheson, D.I.: Relaxation Runge–Kutta methods: Conservation and stability for inner-product norms. SIAM Journal on Numerical Analysis **57**(6), 2850–2870 (2019)
35. Klose, B.F., Jacobs, G.B., Kopriva, D.A.: On the robustness and accuracy of marginally resolved discontinuous Galerkin schemes for two dimensional Navier–Stokes flows. In: AIAA Scitech 2019 Forum, p. 0780. American Institute of Aeronautics and Astronautics (2019)
36. Knepley, M.G., Karpeev, D.A.: Mesh algorithms for PDE with Sieve I: Mesh distribution. Scientific Programming **17**(3), 215–230 (2009)
37. Munson, B.R., Young, B.F., Okiishi, T.H.: Fundamental of fluid mechanics, second edn. Wiley (1990)
38. Nordström, J., Carpenter, M.H.: Boundary and interface conditions for high-order finite-difference methods applied to the Euler and Navier–Stokes equations. Journal of Computational Physics **148**(2), 621–645 (1999)
39. Nordström, J., Carpenter, M.H.: High-order finite-difference methods, multidimensional linear problems, and curvilinear coordinates. Journal of Computational Physics **173**(1), 149–174 (2001)
40. Olsson, P., Oliger, J.: Energy and maximum norm estimates for nonlinear conservation laws. Tech. Rep. 94–01, The Research Institute of Advanced Computer Science (1994)
41. Parsani, M., Boukharfane, R., Nolasco, I., Del Rey Fernández, D.C., Zampini, S., Dalcin, L.: Unveiling the potential of high-order accurate entropy stable discontinuous collocated Galerkin methods for the next generation of compressible CFD frameworks: SSDC algorithms and flow solver. submitted to Journal of Computational Physics (2020)
42. Parsani, M., Carpenter, M.H., Fisher, T.C., Nielsen, E.J.: Entropy stable staggered grid discontinuous spectral collocation methods of any order for the compressible Navier–Stokes equations. SIAM Journal on Scientific Computing **38**(5), A3129–A3162 (2016)
43. Parsani, M., Carpenter, M.H., Nielsen, E.J.: Entropy stable discontinuous interfaces coupling for the three-dimensional compressible Navier–Stokes equations. Journal of Computational Physics **290**, 132–138 (2015)
44. Parsani, M., Carpenter, M.H., Nielsen, E.J.: Entropy stable wall boundary conditions for the three-dimensional compressible Navier–Stokes equations. Journal of Computational Physics **292**(1), 88–113 (2015)
45. Pazner, W., Persson, P.O.: Analysis and entropy stability of the line-based discontinuous Galerkin method. Journal of Scientific Computing **80**(1), 376–402 (2019)
46. Ranocha, H., Sayyari, M., Dalcin, L., Parsani, M., Ketcheson, D.I.: Relaxation Runge–Kutta methods: Fully-discrete explicit entropy-stable schemes for the Euler and Navier–Stokes equations. In Press, SIAM Journal on Scientific Computing (2019)
47. Rojas, D., Boukharfane, R., Dalcin, L., Del Rey Fernández, D.C., Ranocha, H., Keyes, D., Parsani, M.: On the robustness and performance of entropy stable discontinuous collocation methods for the compressible Navier–Stokes equations. submitted to Journal of Computational Physics (2019)
48. Sandham, N.D., Li, Q., Yee, H.C.: Entropy splitting for high-order numerical simulation of compressible turbulence. Journal of Computational Physics **178**(2), 307–322 (2002)
49. Sjörn, B., Yee, H.C.: High order entropy conservative central schemes for wide ranges of compressible gas dynamics and MHD flows. Journal of Computational Physics **364**, 153–185 (2018)
50. Söderlind, G.: Digital filters in adaptive time-stepping. ACM Transactions on Mathematical Software **29**(1), 1–26 (2003)
51. Söderlind, G., Wang, L.: Adaptive time-stepping and computational stability. Journal of Computational and Applied Mathematics **185**(2), 225–243 (2006)
52. Svärd, M.: Weak solutions and convergent numerical schemes of modified compressible Navier–Stokes equations. Journal of Computational Physics **288**(C), 19–51 (2015)
53. Svärd, M., Carpenter, M.H., Parsani, M.: Entropy stability and the no-slip wall boundary condition. SIAM Journal on Numerical Analysis **56**(1), 256–273 (2018)
54. Svärd, M., Nordström, J.: Review of summation-by-parts schemes for initial-boundary-value-problems. Journal of Computational Physics **268**(1), 17–38 (2014)

55. Svärd, M., Özcan, H.: Entropy-stable schemes for the Euler equations with far-field and wall boundary conditions. Journal of Scientific Computing **58**(1), 61–89 (2014)
56. Tadmor, E.: The numerical viscosity of entropy stable schemes for systems of conservation laws I. Mathematics of Computation **49**(179), 91–103 (1987)
57. Tadmor, E.: Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. Acta Numerica **12**, 451–512 (2003)
58. Thomas, D., Lombard, C.K.: Geometric conservation law and its application to flow computations on moving grids. AIAA Journal **17**(10), 1030–1037 (1979)
59. Vinokur, M., Yee, H.C.: Extension of efficient low dissipation high order schemes for 3-d curvilinear moving grids. In: D.A. Caughey, M. Hafez (eds.) Frontiers of Computational Fluid Dynamics, pp. 129–164. World Scientific Publishing Company (2002)
60. de Wiart, C., Hillewaert, K., Duponcheel, M., Winckelmans, G.: Assessment of a discontinuous Galerkin method for the simulation of vortical flows at high Reynolds number. International Journal for Numerical Methods in Fluids **74**(7), 469–493 (2014)
61. Winters, A.R., Derigs, D., Gassner, G.J., Walch, S.: Uniquely defined entropy stable matrix dissipation operator for high Mach number ideal MHD and compressible Euler simulations. Journal of Computational Physics **332**(1), 274–289 (2017)
62. Winters, A.R., J.Gassner, G.: A comparison of two entropy stable discontinuous Galerkin spectral element approximations to the shallow water equations with non-constant topography. Journal of Computational Physics **301**(1), 357–376 (2015)
63. Winters, A.R., Moura, R.C., Mengaldo, G., Gassner, G.J., Walch, S., Peiro, J., Sherwin, S.J.: A comparative study on polynomial dealiasing and split form discontinuous Galerkin schemes for under-resolved turbulence computations. Journal of Computational Physics **372**, 1–21 (2018)
64. Yee, H.C., Vinokur, M., Djomehri, M.J.: Entropy splitting and numerical dissipation. Journal of Computational Physics **162**(1), 33–81 (2000)

A Implementation details for the discretization of the convection-diffusion equation

The expanded version of the discretization in Eqs. (26), (27), (28), and (29) gives further insight into the implementation of the algorithm. However, the compact formulation relies on the use of tensor-products and notational abstractions which can obscure how to implement the algorithm in practice. In this appendix, we try and expand the algorithm in pseudocode to help the interested reader. We present a simple approach to implementing the algorithm, where we make no claim to computational efficiency. Indeed, this is a difficult task because efficiency, particularly in the context of high-performance computing, relies on numerous factors. Nevertheless, we hope that what is presented in this appendix is helpful in obtaining a first version of the algorithm.

Assuming that the metric terms have already been computed, the steps in constructing a right-hand side evaluation for time advancement are as follows:

1. Construct and store the auxiliary flux variables $\boldsymbol{\theta}_a$
   - Compute the on element derivative $\mathsf{D}_{\hat{\xi}_a}\boldsymbol{u}$
   - Compute the SATs
2. Compute the on element inviscid derivative $\sum_{l=1}^{3}\left(\mathsf{D}_{\hat{\xi}_l}\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_l}{\partial x_m}\right]+\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_l}{\partial x_m}\right]\mathsf{D}_{\hat{\xi}_l}\right)\boldsymbol{u}$ for each of the Cartesian directions
3. Compute the on element viscous derivative $\sum_{l=1}^{3}\left(\mathsf{D}_{\hat{\xi}_l}\left[\hat{\mathsf{C}}_{l,a}\right]\boldsymbol{\theta}_a\right)\boldsymbol{u}$ for each of the Cartesian directions
4. Compute the inviscid SATs
5. Compute the viscous SATs

Of course, one can combine the two on element computations and similarly the computation of the SATs.

While tensor-products are used to describe the SBP operators, storing and applying the resulting sparse matrices is not computationally advisable. In a general code that handles arbitrary SBP operators, the SBP matrices could be stored in compressed sparse row format.

However, such a choice has important storage and computational implications. Instead, one can take advantage of the tensor-product nature and apply the operators node by node. For example, the following code applies an SBP operator in a specified direction:

```
function du = derivative(u,D,N,direction)
% u = solution vector organized with xi_3 varying most rapidly,
%       followed by xi_2, and then xi_1
% D = one-dimensional SBP derivative operator
% N = number of nodes in each direction
% direction = 1,2,3 for xi_1, xi_2, xi_3

du = zeros(N^3,1);

if(direction==1)
  cnt = 1;
  stride = N^2;
  for i = 1:N
    for j = 1:N
      for k = 1:N
        for c = 1:N
          du(cnt,1) = du(cnt,1)+D(i,c)*u(cnt+(c-i)*stride);

        end
        cnt = cnt+1;
      end
    end
  end
elseif(direction==2)
  cnt = 1;
  stride = N;
  for i = 1:N
    for j = 1:N
      for k = 1:N
        for c = 1:N
          du(cnt,1) = du(cnt,1)+D(j,c)*u(cnt+(c-j)*stride);
        end
        cnt = cnt+1;
     end
    end
  end
elseif(direction==3)
  cnt = 1;
  stride = 1;
  for i = 1:N
    for j = 1:N
      for k = 1:N
        for c = 1:N
          du(cnt,1) = du(cnt,1)+D(k,c)*u(cnt+(c-k)*stride);
        end
        cnt = cnt+1;
     end
    end
  end
end
```

For the first step, we compute the on element contribution of the viscous fluxes $\mathsf{D}_{\hat{\xi}_a}\boldsymbol{u}$ using the `derivative` function. Next, we apply the SATs, which are constructed from on element face and off element face contributions. However, before presenting the generic code, we take some time to decompose the conforming and nonconforming viscous SATs. The conforming SAT for the L element abutting the R element is constructed as

$$\boldsymbol{SAT}_{\mathrm{L}} \equiv -\frac{1}{2}\mathsf{P}^{-1}\left\{\left(\boldsymbol{e}_N\left(\boldsymbol{e}_N\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}\otimes\mathsf{P}^{(1D)}\right)\boldsymbol{u}_{\mathrm{L}}-\left(\boldsymbol{e}_N\left(\boldsymbol{e}_1\right)^{\mathrm{T}}\otimes\mathsf{P}^{(1D)}\otimes\mathsf{P}^{(1D)}\right)\boldsymbol{u}_{\mathrm{R}}\right\}. \tag{75}$$

Manipulating the tensor products gives

$$\boldsymbol{SAT}_{\mathrm{L}} = \frac{1}{2}\left\{\left(\mathsf{P}^{(1D)}\right)^{-1}\boldsymbol{e}_N\otimes\mathsf{I}_N\otimes\mathsf{I}_N\right\}\left\{\left(\boldsymbol{e}_N^{\mathrm{T}}\otimes\mathsf{I}_N\otimes\mathsf{I}_N\right)\boldsymbol{u}_{\mathrm{L}}-\left(\boldsymbol{e}_1^{\mathrm{T}}\otimes\mathsf{I}_N\otimes\mathsf{I}_N\right)\boldsymbol{u}_{\mathrm{R}}\right\}. \tag{76}$$

The action of $\left(\boldsymbol{e}_N^{\mathrm{T}} \otimes \mathsf{I}_N \otimes \mathsf{I}_N\right)$ on $\boldsymbol{u}_{\mathrm{L}}$ is to construct a vector of size $N \times N$ of the values of $\boldsymbol{u}_{\mathrm{L}}$ at the nodes of the joint surface. Similarly, the action of $\left(\boldsymbol{e}_1^{\mathrm{T}} \otimes \mathsf{I}_N \otimes \mathsf{I}_N\right)$ is to construct a vector of size $N \times N$ of $\boldsymbol{u}_{\mathrm{R}}$ a the nodes of the joint surface. The difference is then computed and this is inserted back onto the appropriate locations in the right-hand side vector, scaled by $1/\mathsf{P}^{(1D)}(N,N)$ by $\left\{\left(\mathsf{P}^{(1D)}\right)^{-1} \boldsymbol{e}_N \otimes \mathsf{I}_N \otimes \mathsf{I}_N\right\}$.

We now analyze the nonconforming SAT in the same way; we reproduce here for convenience the SAT given in the body of the paper:

$$\begin{aligned}\boldsymbol{SAT}_{\mathrm{L}} \equiv -\frac{1}{2}\mathsf{P}_{\mathrm{L}}^{-1}\Bigg\{&\left(\boldsymbol{e}_N^{\mathrm{L}}\left(\boldsymbol{e}_N^{\mathrm{L}}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\right)\boldsymbol{u}_{\mathrm{L}} \\ &- \sum_{f=1}^{4}\left(\boldsymbol{e}_N^{\mathrm{L}}\left(\boldsymbol{e}_1^{\mathrm{R}_f}\right)^{\mathrm{T}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^2_{\mathrm{R}_f\mathrm{toL}} \otimes \mathsf{P}_{\mathrm{L}}^{(1D)}\mathsf{I}^3_{\mathrm{R}_f\mathrm{toL}}\right)\boldsymbol{u}_{\mathrm{R}_f}\Bigg\}.\end{aligned} \tag{77}$$

Performing similar manipulations as before gives

$$\begin{aligned}\boldsymbol{SAT}_{\mathrm{L}} = &\frac{1}{2}\left\{\left(\mathsf{P}_{\mathrm{L}}^{(1D)}\right)^{-1}\boldsymbol{e}_N^{\mathrm{L}} \otimes \mathsf{I}_N^{\mathrm{L}} \otimes \mathsf{I}_N^{\mathrm{L}}\right\} \\ &\left\{\left(\left(\boldsymbol{e}_N^{\mathrm{L}}\right)^{\mathrm{T}} \otimes \mathsf{I}_N^{\mathrm{L}} \otimes \mathsf{I}_N^{\mathrm{L}}\right)\boldsymbol{u}_{\mathrm{L}} - \sum_{f=1}^{4}\left(\left(\boldsymbol{e}_1^{\mathrm{R}_f}\right)^{\mathrm{T}} \otimes \mathsf{I}^2_{\mathrm{R}_f\mathrm{toL}} \otimes \mathsf{I}^3_{\mathrm{R}_f\mathrm{toL}}\right)\boldsymbol{u}_{\mathrm{R}_f}\right\}.\end{aligned} \tag{78}$$

The only matrix that has changed is the one acting on $\boldsymbol{u}_{\mathrm{R}_f}$, i.e., $\left(\left(\boldsymbol{e}_1^{\mathrm{R}_f}\right)^{\mathrm{T}} \otimes \mathsf{I}^2_{\mathrm{R}_f\mathrm{toL}} \otimes \mathsf{I}^3_{\mathrm{R}_f\mathrm{toL}}\right)$. Loosely speaking, this matrix interpolates $\boldsymbol{u}_{\mathrm{R}_f}$ to the face nodes of element L (it does not exactly interpolate, rather the combination of the 4 operators interpolates a function from the face of the macro element to the face of the L element).

With this background, the SAT can be generically viewed as creating a flux difference between the on element face flux and an off element numerical face flux, which is then scaled by the norm matrix and positioned in the appropriate locations in the right-hand side vector. Thus, a generic code for computing these SATs contributions is

```
sat = function sat_scalar(f_on,f_num,P,N,face)
% f_on = is the on element face flux
% f_num = is the numerical face flux composed of the off element flux and possibly the
%         on element flux
% P = is the one-dimensional SBP norm matrix
% N = number of nodes in each direction for the element where f_on is coming from
% face = 1,2,3,4,5,6 where
%        1 = face where xi_1 is minimum
%        2 = face where xi_1 is maximum
%        3 = face where xi_2 is minimum
%        4 = face where xi_2 is maximum
%        5 = face where xi_3 is minimum
%        6 = face where xi_3 is maximum
%  Note that this constructs the difference and appropriately scales it
%  with the norm matrix. You then have to place the result in the appropriate
%  locations in your right-hand side vector.

df = f_on-f_num;

%-set the negative of the component of the outwardfacing normal
if((face==1)||(face==3)||(face==5))
  nxi_neg = 1;
  pinv = P(1,1);
  else
  nxi_neg = -1;
  pinv = P(N,N);
end

sat = nxi_neg*pinv*df;
```

Thus, for the conforming case, in matrix nomenclature what is being passed is

$$\boldsymbol{f}_{on} = \left(\left(\boldsymbol{e}_N^{\mathrm{L}} \right)^{\mathrm{T}} \otimes \mathsf{I}_N^{\mathrm{L}} \otimes \mathsf{I}_N^{\mathrm{L}} \right) \boldsymbol{u}_{\mathrm{L}}, \qquad \boldsymbol{f}_{num} = \frac{1}{2} \left(\boldsymbol{f}_{on} + \left(\boldsymbol{e}_1^{\mathrm{T}} \otimes \mathsf{I}_N \otimes \mathsf{I}_N \right) \boldsymbol{u}_{\mathrm{R}} \right).$$

In the nonconforming case, the only thing that changes is $\boldsymbol{f}_{num}$, which is given as

$$\boldsymbol{f}_{num} = \frac{1}{2} \left(\boldsymbol{f}_{on} + \sum_{f=1}^{4} \left(\left(\boldsymbol{e}_1^{\mathrm{R}_f} \right)^{\mathrm{T}} \otimes \mathsf{I}^2_{\mathrm{R}_f \mathrm{toL}} \otimes \mathsf{I}^3_{\mathrm{R}_f \mathrm{toL}} \right) \boldsymbol{u}_{\mathrm{R}_f} \right).$$

Next, the inviscid derivatives are computed. As for the viscous fluxes computation, we can construct an approach that does not use matrix multiplication of the sparse matrices $\mathsf{D}_{\hat{\xi}_l}$ and $\left[\hat{\mathcal{J}}_\kappa \frac{\partial \hat{\xi}_l}{\partial x_m} \right]$. The following function performs the required operations:

```
function du = derivative_curv(u,Jdxidx,D,N,m)
% u = solution vector organized with xi_3 varying most rapidly,
%       followed by xi_2, and then xi_1
% Jdxidx = array orgnaized as (xi_1,xi_2,xi_3,1:9), where
% Jdxidx(i,j,k,1) = Jdxi_1/dx_1, Jdxidx(i,j,k,2) = Jdxi_1/dx_2,
% Jdxidx(i,j,k,3) = Jdxi_1/dx_3
% Jdxidx(i,j,k,4) = Jdxi_2/dx_1, Jdxidx(i,j,k,5) = Jdxi_2/dx_2,
% Jdxidx(i,j,k,6) = Jdxi_2/dx_3
% Jdxidx(i,j,k,7) = Jdxi_3/dx_1, Jdxidx(i,j,k,8) = Jdxi_3/dx_2,
% Jdxidx(i,j,k,9) = Jdxi_3/dx_3

% D = one-dimensional SBP derivative operator
% N = number of nodes in each direction
% m = Cartesian coordinate direction 1,2,3 for x_1, x_2, x_3

if(m==1)
  d_xi_1 = 1;% which metric in the xi_1 direction
  d_xi_2 = 4;% which metric in the xi_2 direction
  d_xi_3 = 7;% which metric in the xi_3 direction
elseif(m==2)
  d_xi_1 = 2;
  d_xi_2 = 5;
  d_xi_3 = 8;
elseif(m==3)
  d_xi_1 = 3;
  d_xi_2 = 6;
  d_xi_3 = 9;
end
du = zeros(N^3,1);

cnt = 1;
s_xi_1 = N^2;% stride in the xi_1 direction
s_xi_2 = N  ;% stride in the xi_2 direction
s_xi_3 = 1  ;% stride in the xi_3 direction
for i = 1:N
  for j = 1:N
    for k = 1:N
      for c = 1:N
        du(cnt) = du(cnt)+...
        0.5*D(i,c)*u(cnt+(c-i)*s_xi_1,1)*(Jdxidx(i,j,k,d_xi_1)+...
                                          Jdxidx(c,j,k,d_xi_1)    )+...
        0.5*D(j,c)*u(cnt+(c-j)*s_xi_2,1)*(Jdxidx(i,j,k,d_xi_2)+...
                                          Jdxidx(i,c,k,d_xi_2)    )+...
        0.5*D(k,c)*u(cnt+(c-k)*s_xi_3,1)*(Jdxidx(i,j,k,d_xi_3)+...
                                          Jdxidx(i,j,c,d_xi_3)    );
      end
      cnt = cnt+1;
    end
  end
end
```

The remaining computations can be carried out by reusing the above functions. Thus, the viscous fluxes can be computed using the `derivative` function, where the $\boldsymbol{\theta}_a$ fluxes need to be

scaled by $\left[\hat{\mathsf{C}}_{l,a}\right]$. Moreover, the inviscid and viscous SATs can be computed using the `sat_scalar` function with the appropriate construction of `f_on` and `f_num`.

B Implementation details for the entropy conservative discretization

The viscous terms follow directly from the discretization of the viscous terms in the convection-diffusion equation. However, the introduction of the Hadamard product in the inviscid terms requires further explanation.

We start first with the derivative operator and as in the convection-diffusion context we do not implement the operator by multiplying out matrices and taking the Hadamard product. Instead, we proceed pointwise. The following code illustrates how to compute

$$\sum_{l=1}^{3}\left(\mathsf{D}_{\hat{\xi}_l}\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_l}{\partial x_m}\right]+\left[\hat{\mathcal{J}}_\kappa\frac{\partial\hat{\xi}_l}{\partial x_m}\right]\mathsf{D}_{\hat{\xi}_l}\right)\circ\mathsf{F}_{x_m}\left(\boldsymbol{q},\boldsymbol{q}\right)\mathbf{1}.$$

```
function dq = derivative_curv_Hadamard(q,Jdxidx,D,N,m)
% q = solution vector (vector of conservative variables)
% organized with xi_3 varying most rapidly, followed by
% xi_2, and then xi_1 and is of size (N^3,5)
% Jdxidx = array orgnaized as (xi_1,xi_2,xi_3,1:9), where
% Jdxidx(i,j,k,1) = Jdxi_1/dx_1, Jdxidx(i,j,k,2) = Jdxi_1/dx_2,
% Jdxidx(i,j,k,3) = Jdxi_1/dx_3
% Jdxidx(i,j,k,4) = Jdxi_2/dx_1, Jdxidx(i,j,k,5) = Jdxi_2/dx_2,
% Jdxidx(i,j,k,6) = Jdxi_2/dx_3
% Jdxidx(i,j,k,7) = Jdxi_3/dx_1, Jdxidx(i,j,k,8) = Jdxi_3/dx_2,
% Jdxidx(i,j,k,9) = Jdxi_3/dx_3

% D = one-dimensional SBP derivative operator
% N = number of nodes in each direction
% m = Cartesian coordinate direction 1,2,3 for x_1, x_2, x_3
% Note that flux_fun is a function that calculates the two-point flux
% functions using, for example, the Chandrashekar flux function in the
% m Cartesian direction

if(m==1)
  d_xi_1 = 1;% which metric in the xi_1 direction
  d_xi_2 = 4;% which metric in the xi_2 direction
  d_xi_3 = 7;% which metric in the xi_3 direction
elseif(m==2)
  d_xi_1 = 2;
  d_xi_2 = 5;
  d_xi_3 = 8;
elseif(m==3)
  d_xi_1 = 3;
  d_xi_2 = 6;
  d_xi_3 = 9;
end
dq = zeros(N^3,1);

cnt = 1;
s_xi_1 = N^2;% stride in the xi_1 direction
s_xi_2 = N  ;% stride in the xi_2 direction
s_xi_3 = 1  ;% stride in the xi_3 direction
for i = 1:N
  for j = 1:N
    for k = 1:N
      for c = 1:N
        flux1 = flux_fun(q(cnt,:),q(cnt+(c-i)*s_xi_1,:),m);
        flux2 = flux_fun(q(cnt,:),q(cnt+(c-j)*s_xi_2,:),m);
        flux3 = flux_fun(q(cnt,:),q(cnt+(c-k)*s_xi_3,:),m);
        dq(cnt,1) = dq(cnt,1)+...
        D(i,c)*flux1*(Jdxidx(i,j,k,d_xi_1)+Jdxidx(c,j,k,d_xi_1))+...
        D(j,c)*flux2*(Jdxidx(i,j,k,d_xi_2)+Jdxidx(i,c,k,d_xi_2))+...
        D(k,c)*flux3*(Jdxidx(i,j,k,d_xi_3)+Jdxidx(i,j,c,d_xi_3));
```

```
      end
      cnt = cnt+1;
    end
  end
end
```

Note that as stated in the code, the function `flux_fun` needs to be furnished. This is the function that computes the two point-fluxes and can be constructed, for example, using the approach of Chandrashekar [13].

We will reuse the SAT routine used in the convection-diffusion discretization. In order to see how this can be done, consider first that the Hadamard product between two matrices is the pointwise multiplication of the entries of the matrices. This means that the sparsity pattern of the left matrix must be preserved (and of course any additional sparsity in the right matrix). Examining the matrices involved in the SATs leads to the conclusion that only the solution at the nodes of the elements on either side of the interface are involved. Further careful examination of the matrices involved in the SAT calculation and taking advantage of the property of Hadamard products that $\mathsf{C}\left(\mathsf{A}\circ\mathsf{B}\right)=\left(\mathsf{CA}\right)\circ\mathsf{B}$, if C is diagonal, leads to the conclusion that the SAT, for example on element L, can be rearranged as

$$\begin{aligned}\boldsymbol{SAT}_{\mathrm{L}} = &\left(\left(\mathsf{P}_{\mathrm{L}}^{(1D)}\boldsymbol{e}_{N}^{\mathrm{L}}\right)^{-1}\otimes\mathsf{I}_{N}^{\mathrm{L}}\otimes\mathsf{I}_{N}^{\mathrm{L}}\otimes\mathsf{I}_{5}\right)\\ &\sum_{l=1}^{3}\left\{\mathcal{F}_{x_m}\left(\boldsymbol{u}_{\mathrm{L}}^{\Gamma}\right)-\sum_{f=1}^{4}\left(\mathsf{I}_{\mathrm{R}_f\mathrm{toL}}^{2}\otimes\mathsf{I}_{\mathrm{R}_f\mathrm{toL}}^{3}\otimes\mathsf{I}_{f}\right)\circ\mathsf{F}_{x_m}\left(\boldsymbol{q}_{\mathrm{L}}^{\Gamma},\boldsymbol{q}_{\mathrm{R}_f}^{\Gamma}\right)\mathbf{1}_{\mathrm{R}_f}^{\Gamma}\right\}.\end{aligned}\tag{79}$$

The vector $\mathbf{1}_{\mathrm{R}_f}^{\Gamma}$ is a vector of ones of size $\left(N^{\mathrm{R}_f}\right)^2\times 5$ and

$$\boldsymbol{q}_{\mathrm{L}}^{\Gamma}\equiv\left(\left(\boldsymbol{e}_{N}^{\mathrm{L}}\right)^{\mathrm{T}}\circ\mathsf{I}_{N}^{\mathrm{L}}\otimes\mathsf{I}_{N}^{\mathrm{L}}\otimes\mathsf{I}_{5}\right)\boldsymbol{q}_{\mathrm{L}},\qquad \boldsymbol{q}_{\mathrm{R}_f}^{\Gamma}\equiv\left(\left(\boldsymbol{e}_{1}^{\mathrm{R}_f}\right)^{\mathrm{T}}\circ\mathsf{I}_{N}^{\mathrm{R}_f}\otimes\mathsf{I}_{N}^{\mathrm{R}_f}\otimes\mathsf{I}_{5}\right)\boldsymbol{q}_{\mathrm{R}_f},$$

i.e., $\boldsymbol{q}_{\mathrm{L}}^{\Gamma}$ and $\boldsymbol{q}_{\mathrm{R}_f}^{\Gamma}$ are vectors that contain the solution at the nodes of the interface of the L and R_f elements, respectively.

Now, we can see that the SAT is nothing more than a difference between two fluxes. We reiterate that the action of $\left(\left(\mathsf{P}_{\mathrm{L}}^{(1D)}\boldsymbol{e}_{N}^{\mathrm{L}}\right)^{-1}\otimes\mathsf{I}_{N}^{\mathrm{L}}\otimes\mathsf{I}_{N}^{\mathrm{L}}\otimes\mathsf{I}_{5}\right)$ is to scale the flux. The only object that is uncommon is $\left(\mathsf{I}_{\mathrm{R}_f\mathrm{toL}}^{2}\otimes\mathsf{I}_{\mathrm{R}_f\mathrm{toL}}^{3}\otimes\mathsf{I}_{f}\right)\circ\mathsf{F}_{x_m}\left(\boldsymbol{u}_{\mathrm{L}}^{\Gamma},\boldsymbol{u}_{\mathrm{R}_f}^{\Gamma}\right)\mathbf{1}_{\mathrm{R}_f}^{\Gamma}$. The code for computing this flux is given below.

```
function flux_on = interpolated_flux_Hadamard(q_on,q_off,Ioff2on1,Ioff2on2,...
N_on,N_off,m)
% q_on, q_off = values of the solution at the interface on and off the
% element, respectively
%       organized with xi_3 varying most rapidly,
%       followed by xi_2, and then xi_1 and is of size (Non^3,5) or (Noff^3,5)
% Ioff2on1, Ioff2on2 = one dimensional interpolation operator in the two
% coordinate directions on the face
% N_on, N_off = number of nodes in each computational direction
% m = Cartesian coordinate direction 1,2,3 for x_1, x_2, x_3
% Note that flux_fun is a function that calculates the two-point flux functions
% using, for example, the Chandrashekar flux function in the m Cartesian
% direction

flux_on = zeros(N_on^2,size(q_on,2));
cnt_on = 1;
for i_on = 1:N_on
  for j_on = 1:N_on
  cnt_off = 1;
    for i_off = 1:N_off
      for j_off = 1:N_off
        num_flux = flux_fun(q_on(cnt_on,:),q_off(cnt_off,:),m);
        flux_on(cnt_on,:) = flux_on(cnt_on,:)+...
        Ioff2on1(i_on,i_off)*Ioff2on2(j_on,j_off)*num_flux;
        cnt_off = cnt_off+1;
```

```
      end
    end
    cnt_on = cnt_on+1;
  end
end
```

Finally, `f_on`= $\boldsymbol{\mathcal{F}}_{x_m}\left(\boldsymbol{q}_{\mathrm{L}}^{\Gamma}\right)$ and `f_num`= $\frac{1}{2}\left(\boldsymbol{\mathcal{F}}_{x_m}\left(\boldsymbol{q}_{\mathrm{L}}^{\Gamma}\right)+\texttt{fstar}\right)$, where `fstar = interpolated_flux_Hadamard(...)`.